# USING APPROXIMATE DYNAMIC PROGRAMMING TO CONTROL AN ELECTRIC VEHICLE CHARGING STATION SYSTEM

by

YING CHEN

Presented to the Faculty of the Graduate School of

The University of Texas at Arlington in Partial Fulfillment

of the Requirements

for the Degree of

DOCTOR OF PHILOSOPHY

THE UNIVERSITY OF TEXAS AT ARLINGTON

JULY 2017

**ACKNOWLEDGEMENTS**

First and foremost, I would like to express my deepest gratitude to my supervising professors, Dr. Victoria Chen and Dr. Jay Rosenberger, for all their care, help and guidance in the past three years. They recruited me as a research assistant to complete a NSF funding project at the beginning time. Approximate dynamic programming is required to be used to solve this complex problem. However, I knew nothing about it at that time. In these three years, I made a lot of mistakes due to the lacking of the related knowledge but they always behaved very professionally and patiently to teach and correct me. As professors, they never push students to quickly generate an idea, quickly complete a work and quickly publish a paper like some other professors. Instead, every time, when I showed the experiment results to them, they always asked me why this was good or why this was bad. In their way, they wanted me to learn the nature of the problems but not publish a paper without fully understanding. With their invaluable mentoring, I completed my Ph.D. study within three years.

Second, I want to acknowledge the wonderful support of my committee members: Dr. Wei-Jen Lee and Dr. Aera LeBoulluec. Every time, when I asked them questions, they always used their professional knowledge to clarify my confusion patiently and nicely.

I also would like to thank my friends' help from Feng Liu and Xinglong Ju. Both of them are very smart and nice guy I have ever met. In addition, my gratitude will be given to my colleagues: Gazi, Khan, Nilabh, Hadis. Thank them for being nice and helpful to me.

Last, I want to give my deep gratefulness to the support from my parents and my elder brother, who live in China. Their teachings continue to make me be a better person.

Aug. 7, 2017

**ABSTRACT**

USING APPROXIMATE DYNAMIC PROGRAMMING TO CONTROL AN ELECTRIC VEHICLE CHARGING STATION SYSTEM

Ying Chen, Ph.D.

The University of Texas at Arlington, 2017

Supervising Professors: Victoria C.P. Chen, Jay M. Rosenberger

Dynamic programming (DP) as a mathematical programming approach to optimize a system evolving over time has been applied to solve the multi-stage optimization problems in a lot of areas such as manufacturing systems and environmental engineering. Due to the "curses of dimensionality", traditional DP method is only able to solve a low dimensional problem or problems under very limiting restrictions, In order to employ DP to solve high-dimensional practical complex systems, approximate dynamic programming (ADP) is proposed. Several versions of ADP has been introduced in the literature and for this study, the author takes advantage of design and analysis of computer experiments (DACE) approach to discretize the state space via design of experiments and build the value function with statistical tools, which is named as DACE based ADP approach. In this research, the author first takes advantage of support vector regression (SVR) to build the value function instead of the previous ones such as neural network and multivariate adaptive regression spines, and explore the performance of SVR in the value function approximation compared to the other techniques. After that, 45-degree line correspondence stopping criterion is specified with an algorithm. Then, we formulates the complex electric vehicle (EV) charging stations system located in Dallas-Fort Worth (DFW) metropolitan area in Texas as a Markov decision process (MDP) problem and DACE based infinite horizon ADP algorithm with SVR is used to solve this high-dimensional, continuous-state, infinite horizon problem. Specified 45-degree line correspondence criterion is used to stop the DP iterations and select the ADP policy. Greedy algorithm as a benchmark is proposed to conduct a comparison through paired t-test with the selected ADP policy. The results demonstrate that DACE based infinite horizon ADP algorithm is able to solve the high-dimensional, large-scale,

complex DP problem over continuous spaces and quantified 45-degree line correspondence rule is able to stop the DP iterations reasonably and select a high-quality ADP policy.

Table of Contents

List of Figures

List of Tables

# Chapter 1

## 1. Introduction

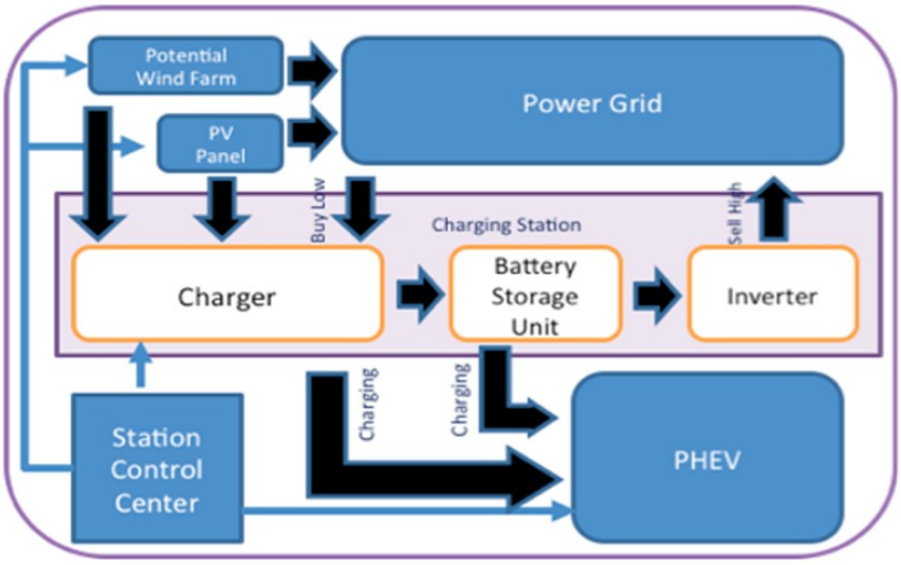
Environmental pollution is becoming more and more serious with the rapid development of our society (Chen et al. 2013). Traditional energy sources such as coal, gas and oil are some of the main sources resulting in pollution, such as greenhouse effects and dust and ashes in the air. Consequently, research on renewable energy sources is becoming more prevalent. With the technology developing, greenhouse gas emissions will be reduced to 17% of 2005 levels by 2020 as the U.S. government pledged (http://www.eia.doe.gov/emeu/aer/pdf/pages/sec12_4.pdf, 2010). After investigation and analysis, the transportation sector, which causes up to 33.1% of all energy related emissions, has been identified as the largest producer of carbon dioxide emission in the U.S. (http://cta.ornl.gov/data/download29.shtml). Therefore, to abate such emissions, electric vehicles have been given a lot of attention. This fundamental transformation from oil based vehicles to electric power ones will help decrease the carbon dioxide emission substantially. Currently, electric vehicles are evolving from hybrid electric vehicle (HEV) to plug-in electric vehicle (PEV) based on the duration of driving and abatement of green house gas emissions. A PEV can be classified into two categories: Plug-in hybrid vehicle (PHEV) and battery electric vehicle (BEV). A PHEV is similar to a HEV, but it has larger battery, and a plug is added to recharge the battery, while a BEV is a purely battery electric vehicle without internal combustion engine, which is in the PHEV. In other words, a BEV is an ideal vehicle that will not emit any carbon dioxide, and a PHEV is an intermediate vehicle between an oil-based vehicle and a BEV. Therefore, the electric vehicle fields are growing very fast. According to Fell et al. (2010), there will be more than 1 million PEVs in the US by 2017. In the DFW metro area, the number of PEVs will increase to around 10,000 by that time.

However, one of the limitations of the development of PEV technology is the location of charging infrastructure. Well planned charging station locations play a critical role in penetrating PEVs into the market. Therefore, different regions have different situations, and in order to support the PEV penetration into local areas, a specific design should be conducted. Furthermore, renewable energy resources, such as wind energy and solar energy, should be integrated into the power grid when designing the charging stations, since they will also contribute to the reduction of greenhouse gas emissions.

### 1.1. Regional PHEV Charging Station Configuration

Before building a charging station for PHEVs, there are several factors that need to be taken in account: 1) market price and wind and solar energy generation vary with time evolving; 2) wind energy and solar energy should be integrated into the power system as sources to provide electricity; 3) charging demand fluctuates based on the number of regional PHEVs; 4) battery size, battery charging and discharging rate in a charging station should be determined; 5) how many charging stations should be built and where they should be located based on the regional population and the quantity of PHEVs; 6) how many charging slots should be open in a charging station. Therefore, regarding these factors, in this research, the objective of the proposed PHEV charging station design is to develop a Level 3 fast DC charging station system equipped with a distributed energy storage system that takes advantage of solar, wind energy, and electricity from the power grid, simultaneously, to charge multiple PHEVs. Regardless of the design part such as the locations of charging station and charging demand based on the local quantity of the PHEVs, we mainly focus on the control part, such as when to buy electricity from the main power grid and when to sell the electricity from the battery back to the main grid. The charging station system will be regarded as an individual, and its objective is to maximize profit through intelligent control. Hence, an operation diagram of a regional PHEV charging station system with $n$ stations is presented in Fig. 1.

In this system, wind energy and solar energy, as renewable energy resources, are integrated to provide electricity. Wind energy is bought as a type of contract, and PV panels are assumed to be placed on the roof of charging stations. Therefore, the cost of purchasing wind energy from remote wind farms is not taken into account in the formulation, and the cost of purchasing the solar panels are also not considered in the control system. From Fig.1, it is observed that the battery as a storage tool plays a critically vital role in this control system. It can store the electricity when the market price is low and sell it back to the power grid for a profit when the market price goes up.




Chapter 1. Figure 1. Regional PEV charging station (Sarikprueck 2015)

For a charging station, its prior job is to satisfy the demand from PHEVs and then to make profits with battery storage. Therefore, when demand arrives at a station, the station control center will make a decision to serve the demand by using direct charge (wind power, PV and the power grid) or battery storage electricity based on the electricity market price.

In this dissertation, we formulate this EV charging station control problem as a Markov decision process (MDP) problem and take advantage of stochastic dynamic programming algorithms to approximate a high-quality solution for the system. In Chapter 2, we will present the literature review about the methodology used in this study; In Chapter 3, the first paper related to methodology of this dissertation is presented; In Chapter 4, based on the methodology introduced in Chapter 3, the second paper about this EV charging station control problem is presented. In Chapter 5, concluding remarks are presented.

**Reference**

Chen, Y., Li, H., Jin, K., Song, Q. Wind farm layout optimization using genetic algorithm with different hub height wind turbines, Energy Conversion and Management, Vol. (70) 56-65, 2013.

Fell, K., Huber, K., Zink, B., Kalisch, R., Forfia, D., Hazelwood, D., Dang, N., Gionet, D., Musto, M., Johnson, W., Assessment of plug-in electric vehicle integration with ISO/RTO systems, *KEMA, Inc. and ISO/RTO Council,* 2010.

Sarikprueck, P. Forecasting Of Wind, PV Generation, And Market Price For The Optimal Operations Of The Regional PEV Charging Stations, Ph.D. dissertation. University of Texas at Arlington, 2015.

## Chapter 2. Support Vector Regression Value Function Approximation for Infinite Horizon Stochastic Dynamic Programming


### Abstract

Approximate dynamic programming (ADP) is a computational approach to provide decision policies for complex dynamic control problems. ADP challenges include high-dimensional and continuous state and decision spaces. A statistical perspective of ADP utilizes design of experiments, to sample a high-dimensional continuous state space, and statistical modeling, to build a continuous value function approximation. In this paper, this statistical perspective is employed with support vector machines (SVM) for value function approximation in an infinite horizon inventory stochastic dynamic programming problem. SVM applications have been successful in a variety of domains, but have not been employed for ADP. Comparisons are made to a prior infinite horizon ADP implementation using multivariate adaptive regression splines (MARS), which have also been used for finite horizon problems. Stopping criteria are discussed, including a 45-degree line correspondence criterion based on a regression concept. SVR is seen to have more stable behavior than MARS. Overall, recommendations are provided to enable good performance using SVR for infinite horizon ADP.



**Keywords**: dynamic programming, design and analysis of computer experiments, support vector regression, stopping criteria


### 1. Introduction

The objective of dynamic programming (DP) is to minimize "cost" or maximize "benefit" of a system evolving over several time periods. For continuous spaces, a typical solution approach discretizes both the state and decision spaces to finite sets. With the increase in computational power, approximate dynamic programming (ADP) methods have grown in popularity, including reinforcement learning (e.g., Sutton and Barto 1998, Castelletti et al. 2010, Wei et al. 2015), neuro-DP (e.g., Bertsekas and Tsitsiklis 1996, Van Roy et al. 1997, Castelletti et al. 2007), and methods using the post-decision state (e.g., Powell 2007, Anderson et al. 2011, Simao et al. 2008). However, high-dimensional problems still face the "curse of dimensionality," which is exponential growth of computational and storage requirements as the dimension of state, decision and/or stochastic variables increase (Powell, 2007).

A statistical perspective enables a more general view of continuous-state DP problems (Chen et al. 1999). This perspective of ADP is analogous to design and analysis of computer experiments (DACE, Chen et al. 2006). State space discretization is based on design of experiments, and value function approximation is based on statistical modeling. Chen et al. (2017) introduced the DACE approach to infinite horizon problems and utilized an experimental design derived from a Sobol low-discrepancy sequence (Sobol 1967) and multivariate adaptive regression splines (MARS), which has previously been used for finite horizon problems (e.g., Chen 1999, Cervellera et al. 2007, Yang et al. 2009) In this paper, we study the use of a support vector regression (SVR, Drucker et al. 1997) within a DACE based infinite horizon ADP algorithm.

Support vector machines (SVM) are discriminative classifiers which formally are defined by a separating hyperplane (Cortes and Vapnik 1995). Initially, the SVM approach was developed for binary classification, which is also called support vector classification (SVC). Later, Drucker et al. (1997) proposed SVR to handle regression-type modeling based on the theory of SVC. In recent years, SVM has been successful in a variety of data mining applications, including healthcare (e.g., Furey et al. 2000, Hua and Sun 2001, Rick et al. 2008), energy (e.g., Mohandes et al. 2004, Sarikprueck et al. 2015, Dong et al. 2005), manufacturing (e.g., Chen and Wang 2007, Martinez-de-Pison et al. 2008, Li and Huang 2009), finance (e.g., Farquad et al. 2012, Yao and Lian 2016, Zhang et al. 2015), etc. To our knowledge, SVR has not been utilized for value function approximation.

In Section 2, we describe the DACE based ADP approach. In Section 3, we compare SVR and MARS using the infinite horizon inventory SDP problem from Chen et al. (2017), which was derived from a prior finite horizon inventory problem (Chen et al. 1999, Chen 1999, Cervellera et al. 2007, Cervellera and Macciò 2011, Cervellera and Macciò 2016) In Section 4, we describe stopping criteria, including a formal stopping rule using a 45-degree line correspondence criterion that was first suggested by Chen et al. (2017). In Section 5, we discuss the computational results using these stopping criteria and other SVR considerations, and in Section 6, we present concluding remarks.

## 2. ADP Approach

In the following, we will first overview the infinite horizon DP formulation (Bellman 1957), then we summarize the DACE based infinite horizon ADP algorithm introduced by Chen et al. (2017).

### 2.1. Infinite Horizon DP Formulation

The future value function (FVF) for an infinite horizon stochastic DP problem can be written as follows:

$$V(x_t) = \min_u E\{\sum_{t=0}^{\infty} \gamma^t c(x_t, u_t, \varepsilon_t)\}, \qquad (1)$$

where $t$ is the time period, $E$ is the conditional expectation under the policy, $u$, $\gamma \in [0,1]$ is a discount factor that handles the tradeoff between the immediate and delayed costs, $x_t$ is the state vector, $c$ is the cost function, $V$ is the FVF, and $\varepsilon_t$ is the stochastic variable. This equation can be written recursively as:

$$V(x_t) = \min_{u_t} \mathrm{E}\{c(x_t, u_t, \varepsilon_t) + \gamma V(x_{t+1})\}$$

$$s.t. \quad x_{t+1} = f(x_t, u_t, \varepsilon_t), \qquad (2)$$

$$(x_t, u_t) \in \Gamma_t,$$

where $f$ is the state transition function and $\Gamma_t$ represents state and decision space constraints. Since in infinite horizon DP, there is only one true value function, a value iteration approach creates a sequence of value functions that eventually converge to the true one (Bertsekas 2017). An ADP version can be written as:

$$\tilde{V}_k(x_t) = \min_{u_t} \mathrm{E}\{c(x_t, u_t, \varepsilon_t) + \gamma \hat{V}_{k-1}(x_{t+1})\}$$

$$s.t. \quad x_{t+1} = f(x_t, u_t, \varepsilon_t), \qquad (3)$$

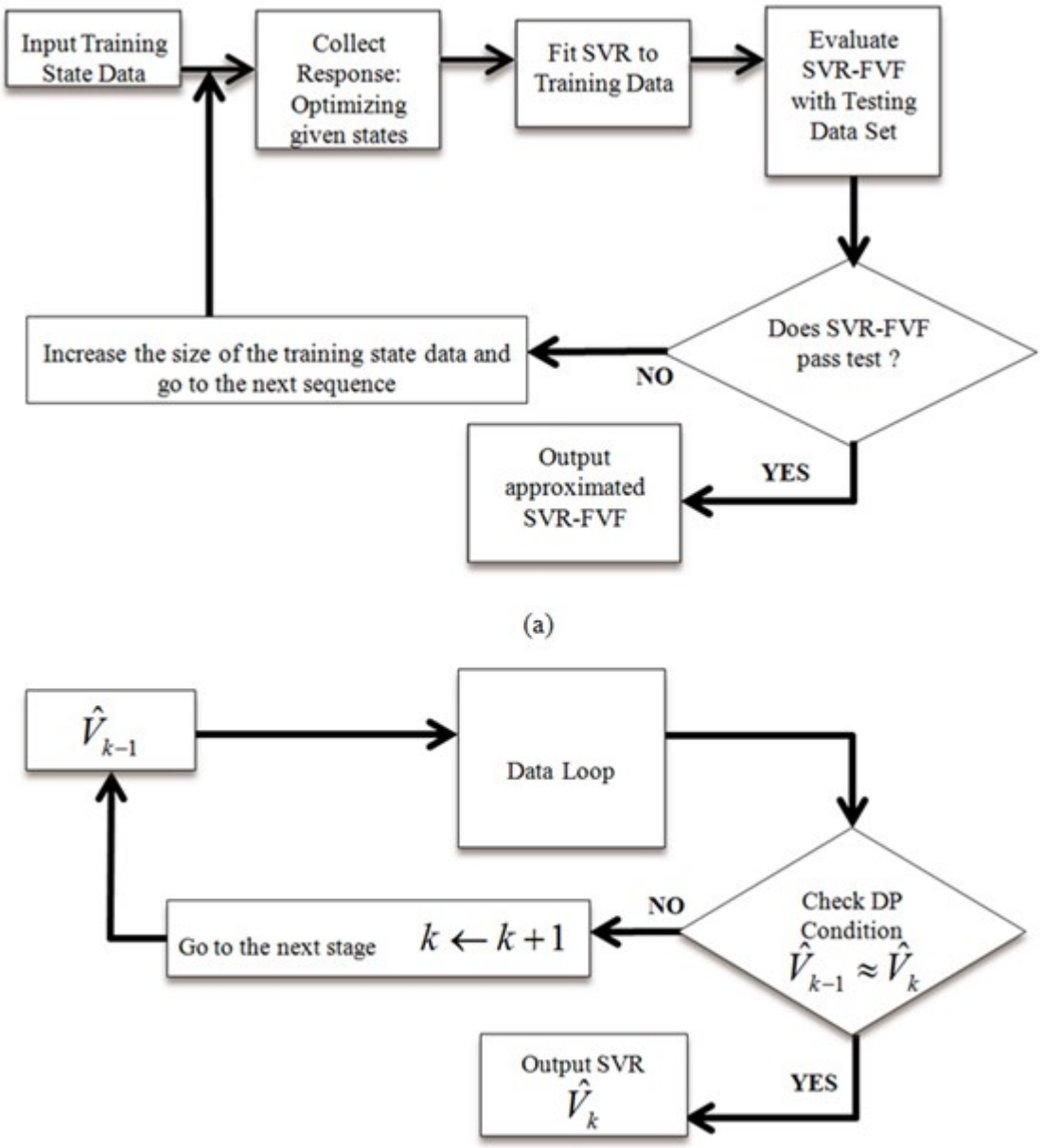
$$(x_t, u_t) \in \Gamma_t,$$

where $\hat{V}_{k-1}$ is the approximate FVF (aFVF) at the $k$-1$^{\text{th}}$ iteration, and the realized $\tilde{V}_k$ values from the minimization are approximated by the aFVF denoted by $\hat{V}_{k-1}$.

### 2.2. DACE Based Infinite Horizon ADP

Chen et al. (2017) used the DACE concept to develop a new algorithm to solve infinite horizon DP problems over a continuous space. In this algorithm, two loops are used to achieve the aFVF: an inner data loop and an outer DP loop. The data loop follows the adaptive value function approximation (AVFA) approach of Fan et al. (2013) to sample the state space sequentially in order to control the amount of sampling needed to build an aFVF. The DP loop follows the value iteration concept to generate a sequence of aFVFs. It should be noted that stopping criteria must be specified in advance for both the data loop and the DP. The flow chart of this algorithm using SVR for the FVF approximation is shown in Fig.1.

For our implementation of SVR, we employed a least squares SVM (LSSVM, Suykens et al. 2002) toolbox in MATLAB (http://www.esat.kuleuven.be/sista/lssvmlab/). The LSSVM algorithm is one of many SVM variants, including linear programming SVM (Zhou et al. 2002), sparse SVM (Bi et al. 2003), etc. These variants differ in their specification of objective functions and constraints. LSSVM is a reformulation of the standard SVM model, which utilizes linear Karush-Kuhn-Tucker conditions (Suykens et al. 2002). LSSVM is also closely related to Gaussian processes and regularization networks, but additionally emphasizes and exploits primal-dual interpretations (Suykens et al. 2002). In the LSSVM toolbox, the Gaussian RBF kernel was selected, which requires adjustment of its two main parameters: bandwidth and the regularization ratio. The function "tunelssvm" in this toolbox was used to tune these two parameters via a grid-search algorithm. Specifically, the tuning procedure consists of two steps: 1) a coupled simulated annealing algorithm to determine the suitable tuning parameter and 2) a simplex method that performs fine tuning of the parameters (Brabanter et al. 2011). This approach enables adaptive tuning of the SVR model, which is necessary for the AVFA approach (Fan et al. 2013) to approximate the value function, as shown in Fig. 1.




Chapter 2. Figure 1. DACE based infinite horizon ADP algorithm (Chen et al. 2017): (a) data loop, (b) DP loop

## 3. Comparison of SVR and MARS

In this section, we describe the infinite horizon inventory stochastic DP problem utilized by Chen et al. (2017), and then present comparisons between SVR and MARS using the DACE based infinite horizon ADP algorithm in Fig. 1.

### 3.1. Infinite Horizon Inventory Stochastic DP Problem

The inventory problem involves a nine-dimensional nearly continuous state space. It takes advantage of forecasts of customer demand with the martingale model of forecast evolution (MMFE, Heath and Jackson 1994) to evolve the state variables over time. Suppose there are $n_I$ different products and forecasts for demand are made 0, 1,…, and ($K$ - 1) months ahead, then there are $n_I$ ($K$ + 1) state variables. The state of the system at time $t$, can be defined as

$$\boldsymbol{x}_t = \left(I_t^{(1)}, \dots, I_t^{(n_I)}, D_{(t,t)}^{(i)}, \dots, D_{(t,t)}^{(n_I)}, \dots, D_{t,t+K-1}^{(n_I)}\right), \qquad (4)$$

where $I_t^{(i)}$ is the inventory level of product $i$ at the beginning of time period $t$ and $D_{(t,t+k)}^{(i)}$ is the forecast determined at the beginning of time period $t$ to predict the demand of product $i$ in time period $t + k$.

The decision vector is $\boldsymbol{u}_t = (u_t^{(1)}, \dots, u_t^{(n_I)})$, where $u_t^{(i)}$ is the amount of product $i$ ordered in period $t$. Let $D_{(t,t+k)} = \left(D_{(t,t+k)}^{(1)}, \dots, D_{(t,t+k)}^{(n_I)}\right)$, and $\boldsymbol{\mu}_t = \left(\mu_t^{(1)}, \dots, \mu_t^{(n_I)}\right)$ be the vector of mean demands, where $\mu_t^{(i)}$ is the mean demand for product $i$ in time period $t$. The mean demand $\mu_t^{(i)}$ is utilized as the initial forecast of demand for product $i$ in time period $t$. In Chen et al. (1999), $n_I$ is set to 3 and $K$ is set to be 2, so the state space dimension is 9. At the beginning of time period $t$, the forecasts in the current period are $D_{(t,t)}$, and the forecasts for the next period are $D_{(t,t+1)}$.

Following MMFE, the state transition model from time period $t$ to $t$ + 1 for each product is:

$$I_{t+1}^{(i)} = I_t^{(i)} + u_t^{(i)} - \left(D_{(t,t)}^{(i)} \cdot \varepsilon_{(t,t)}^{(i)}\right), \qquad (5)$$

$$D_{(t+1,t+1)}^{(i)} = \left(D_{(t,t+1)}^{(i)} \cdot \varepsilon_{(t,t+1)}^{(i)}\right), \qquad (6)$$

$$D_{(t+1,t+2)}^{(i)} = \left(\mu_{t+2}^{(i)} \cdot \varepsilon_{(t,t+2)}^{(i)}\right), \qquad (7)$$

where $\varepsilon^{(i)}_{(t,t+2)}$ represents the multiplicative error in the forecast for time period $t + 2$ from the mean demand for that period, $\varepsilon^{(i)}_{(t,t+1)}$ denotes the multiplicative error in the forecast for the current time period $t$+1 from the forecast made in period $t$, and $\varepsilon^{(i)}_{(t,t)}$ is the multiplicative error in the forecast for the demand in time period $t$. Specifically, the actual demand in period $t$ is modeled as $(D^{(i)}_{(t,t)} \cdot \varepsilon^{(i)}_{(t,t)})$. The multiplicative errors in the forecast for period $t + k$ are:

$$\varepsilon^{(i)}_{(t,t+k)} = \frac{D^{(i)}_{(t+1,t+k)}}{D^{(i)}_{(t,t+k)}}, \qquad (8)$$

and are assumed to have a mean of one, therefore forming a martingale from the sequence of future forecasts for period $t + k$ (Heath and Jackson 1994). Let $\varepsilon_t$ be the $3n_I \times 1$ vector:

$$\varepsilon_t = \left(\varepsilon^{(0)}_{(1,t)}, \ldots, \varepsilon^{(0)}_{(n_I,t)}, \varepsilon^{(1)}_{(1,t)}, \ldots, \varepsilon^{(1)}_{(n_I,t)}, \ldots, \varepsilon^{(2)}_{(n_I,t)}, \ldots, \varepsilon^{(2)}_{(n_I,t)}\right). \quad (9)$$

The random vector $\varepsilon_t$ is assumed to follow a multivariate lognormal distribution (Chen et al. 1999, Heath and Jackson 1994). The standard inventory cost function is V-shaped, involving inventory holding costs and backorder costs, as shown below:

$$c_v(x_t, u_t) = \sum_{i=1}^{3}(h_i \left[I^{(i)}_{t+1}\right]_+ + \pi_i \left[-I^{(i)}_{t+1}\right]_+), \qquad (10)$$

where $h_i$ is the holding cost parameter for product $i$, and $\pi_i$ is the backorder cost parameter for project $i$. A smoothed version of the cost function was used by Chen et al. (2017) (see Chen et al. 1999 for details on the smoothed version).

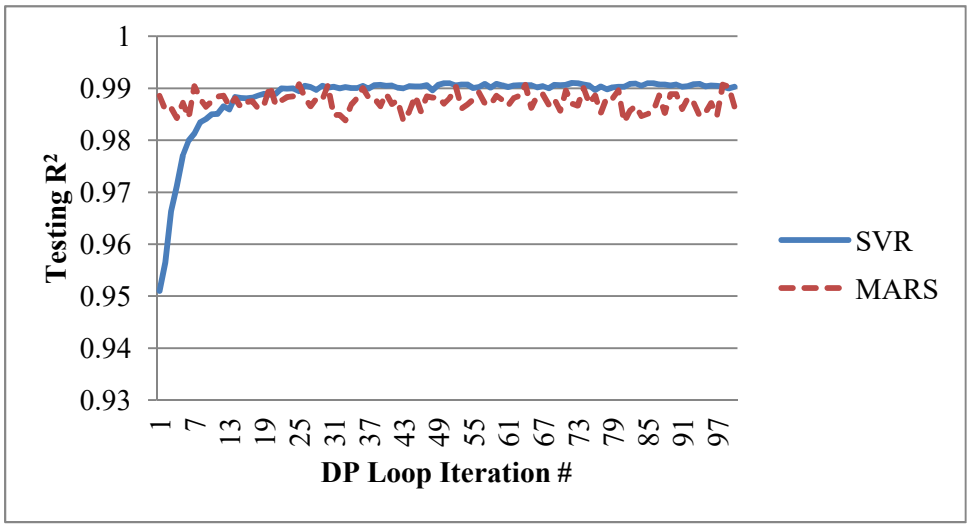
### 3.2. Approximation of aFVFs using SVR

To implement the algorithm in Fig. 1, we utilized the same experimental design process as Chen et al. (2017) and only compared SVR vs. MARS for the FVF approximation. Chen et al. (2017) employed a Sobol's low-discrepancy sequence (Sobol 1967) to sample state points for the training data set, and a Halton's low-discrepancy sequence (Halton 1960) for the testing data set. The range of each of the nine state variables is in Table 2. 1:

Table 2. 1 Range of each variable in inventory forecasting problem

| # of Variable | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 |
|---|---|---|---|---|---|---|---|---|---|
| Min | -20 | -24 | -15 | 0 | 0 | 0 | 0 | 0 | 0 |
| Max | 20 | 24 | 15 | 20 | 24 | 15 | 13 | 16 | 10 |

The first component of the algorithm in Fig. 1 is the data loop. The stopping criterion implemented for the data loop uses the difference between the testing $R^2$ for two consecutive data loops and the value of testing $R^2$. If this difference is less than 0.05 and testing $R^2$ is also greater than 0.8 simultaneously, then the data loop will stop, otherwise, the data loop continues by sampling another 50 state points for training. The $R^2$ metric is used to indicate how well the fitted SVR model predicts the testing data. The initial size of the training data set is 150, and the size of testing data set is 250. Fig. 2 plots the testing $R^2$ from each iteration of the DP loop, up to 100 iterations. The total computational time for these 100 iterations conducted in MATLAB 2016b on a Lenovo computer with a Xeon, 16-core, 2.8 GHz CPU, was 45 minutes. From Fig. 2, it can be seen that after the 21$^{st}$ iteration, the testing $R^2$ rises above 0.99 and is steadier compared to MARS. This demonstrates that SVR yields a more stable approximation than MARS.




Chapter 2. Figure 2. Testing $R^2$ curve as the DP loop iterations increase

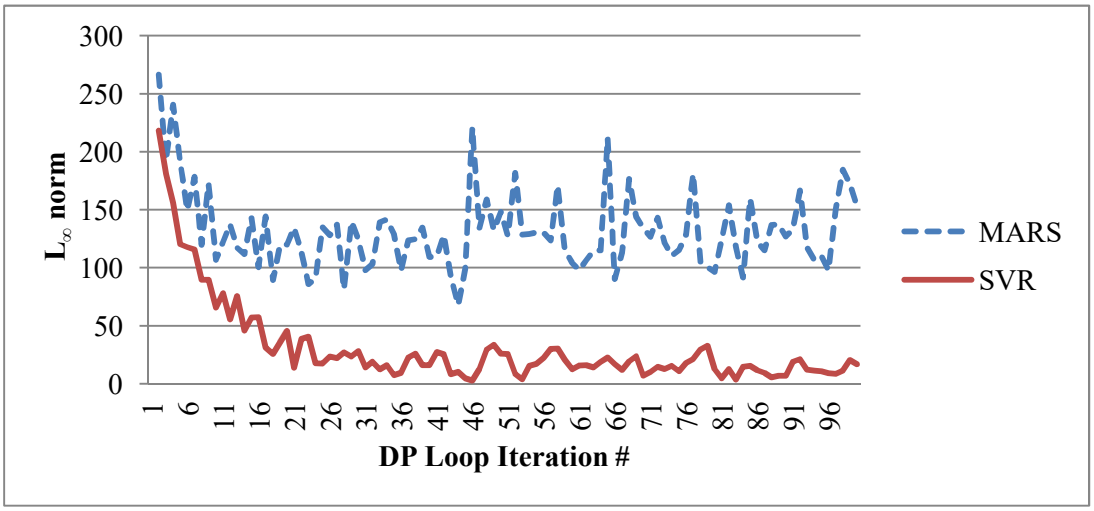
To assess the behavior of the DACE based infinite horizon ADP algorithm using SVR vs. MARS, we computed two stopping criteria. The first is the $L_\infty$ norm (Powell 2007) defined as:

$$\|V_k - V_{k-1}\|\,, \quad \text{where } \|V\| = \max_s |V(s)|. \qquad (11)$$

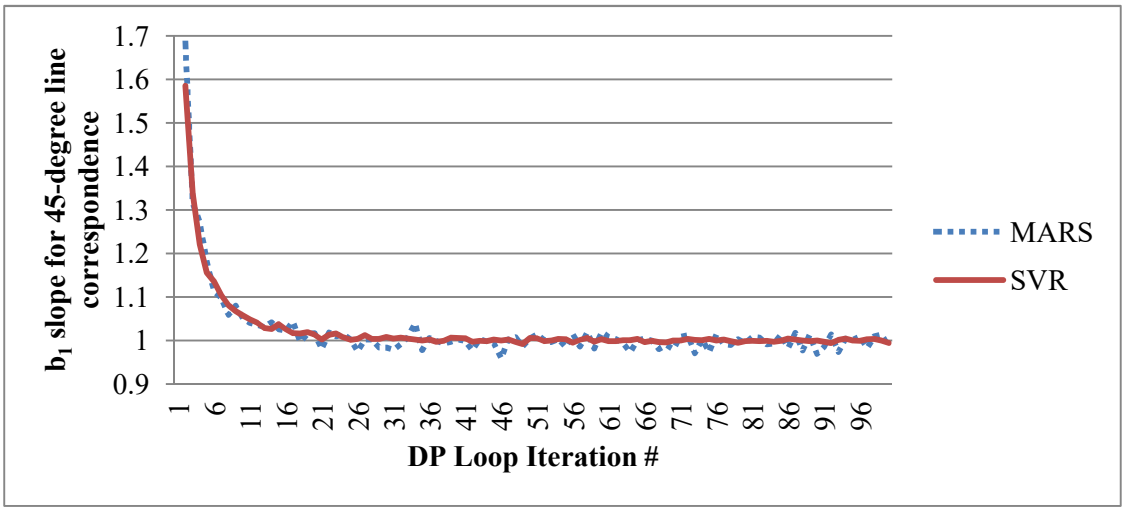
The second uses the estimated slope for 45-degree line correspondence, denoted by $b_1$ in Chen et al. (2017). The 45-degree line correspondence criterion ensures the stability of the shape of value function. For this criterion, a linear regression is fit between two consecutive sets of $\tilde{V}$ values from Eq. (3). The fitted model is specified in Eq. (13):

$$\hat{Y}_k = b_0 + b_1 X_{k-1}\,, \qquad (12)$$

where $X_{k-1}$ is $\tilde{V}$ from iteration $k-1$, $Y_k$ is $\tilde{V}$ from iteration $k$, $\hat{Y}_k$ estimates $Y_k$, $b_0$ estimates the intercept, and $b_1$ estimates the slope. If there is an exact 45-degree line correspondence between the value function data from the two consecutive iterations, then the intercept should be 0, and the slope should be 1. In Figs. 3 and 4, the comparison between MARS and SVR is shown using these two criteria. From Fig. 3, the $L_\infty$ norm with SVR levels off, especially starting from the 24th iteration. For MARS, the $L_\infty$ norm metric was unable to level off within the 100 DP iterations, even with 6000 iterations as shown in Chen et al. (2017). In Fig. 4, the $b_1$ metric with SVR starts to level off around the 21st iteration, and the $b_1$ values from the 21st to 100th iterations are between 0.991 to 1.015, which are very close to the ideal value of 1.0. Compared to the $b_1$ values for MARS, SVR achieves much smaller variation. Overall, based on Figs. 2-4, we can conclude that SVR yields more stable performance than MARS for a DACE based infinite horizon ADP algorithm. In the next section, we formally specify a stopping rule based on the 45-degree line correspondence criterion.



Chapter 2. Figure 3. Variations of $L_\infty$ norm value in the first 100 DP iterations




Chapter 2. Figure 4. Variations of 45-degree line correspondence rule in the first 100 DP iterations.

## 4. Stopping Criteria

The typical stopping criterion for value iteration uses the $L_\infty$ norm proposed by Powell (2017):

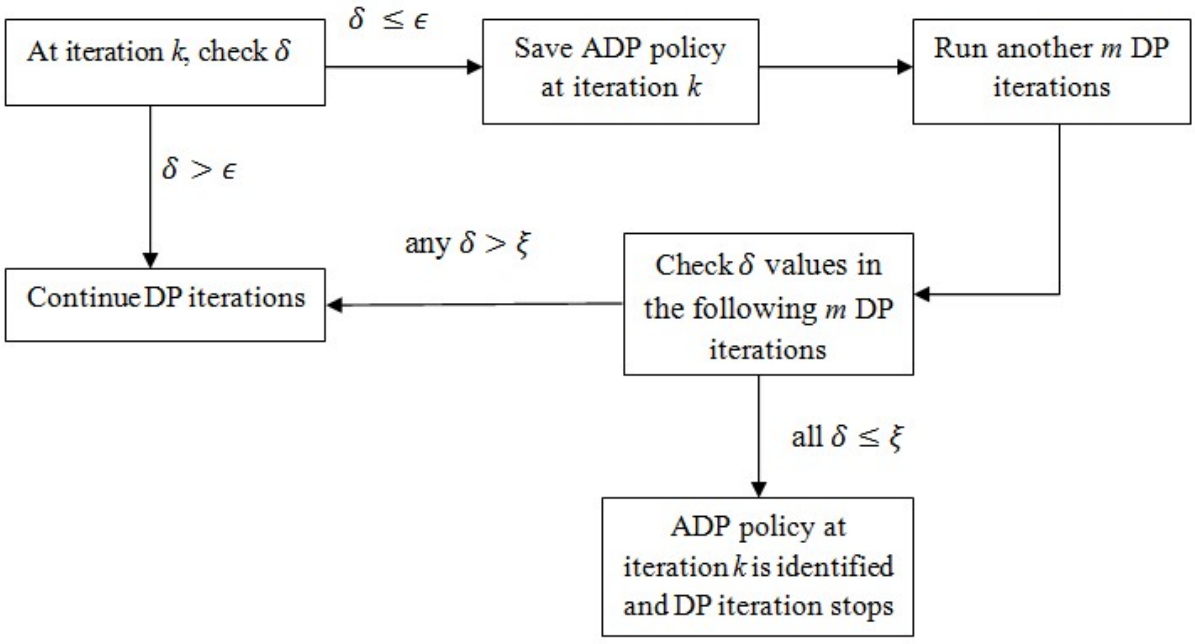
$$\|V_k - V_{k-1}\| < \frac{\theta(1-\gamma)}{2\gamma}. \qquad (13)$$

Thus, the stopping criterion is reached when the maximum change in the value of any state is lower than the setting of right-hand side in Eq. (3), where $\gamma$ is the discount factor, and $\theta$ is a specified error tolerance. In this study, the discount factor is 0.9, which is the same as Chen et al. (2017). Using the 45-degree line correspondence criterion (Chen et al. 2017), the algorithm should stop once the shape of value function has stabilized. As stated by Chen et al. (2017), we only need to pay attention to $b_1$ since $b_0$ only affects the vertical position of the shape. An appropriate stopping rule should identify when $b_1$ has leveled-off and is sufficiently close to 1. While Chen et al. (2017) used $b_1$ to identify high-quality ADP policies, they did not specify a formal stopping rule, so we propose one here to work with the SVR value function approximation.

In Eq. (13), $b_1$ compares consecutive $\tilde{V}$ values from iteration $k$ and iteration $k$ - 1. We refer to this as $b_1$ with 1 lag, denoted as $b_1'$. Alternately, we could calculate $b_1$ between $\tilde{V}$ values at iteration $k$ and iteration $k$ - 2 and refer to this as $b_1$ with 2 lags, denoted as $b_1''$. If $b_1'$ and $b_1''$ are both close to 1, our stopping rule will stop the algorithm. By looking at both lag 1 and lag 2 of the slope estimate $b_1$, we have a longer assessment of the stability of the aFVF. Specifically, our stopping criterion uses the difference between $b_1'$ and $b_1''$:

$$\delta = b_1'' - b_1' \ . \quad (14)$$

Intuitively, the ADP policy at iteration $k$ should be closer to the ADP policy at iteration $k$ - 1 than the ADP policy at iteration $k$ - 2, especially in the early DP iterations. Therefore, $b_1''$ usually is bigger than $b_1'$. However, in later iterations, close to when the algorithm should stop, the two values will start to cross. Once $\delta$ is less than or equal to an error tolerance $\epsilon$, $b_1''$ and $b_1'$ can be considered close enough. However, to ensure that stability has been reached, we save this potential high-quality ADP policy, then continue to run $m$ DP iterations to further identify if this saved ADP policy is sufficient. An additional error tolerance is used as follows: if all $\delta$ values from the next $m$ DP iterations, are less than $\xi$, then this saved ADP policy is finally identified as sufficient, and the algorithm stops. A flowchart of this algorithm is shown in Fig. 5.



Chapter 2. Figure 5. Flowchart of specified 45-degree line correspondence stopping criterion algorithm

## 5. Discussion of Computational Results

Two issues are discussed in this section. First, the two stopping criteria described in Section 4 are used to select ADP policies, and then these policies are simulated to explore how these ADP policies perform. Second, the issue of extrapolation is examined for SVR.

### 5.1. Simulating ADP Policies

First, we need to specify the error tolerances for the two stopping criteria. For the $L_\infty$ norm rule, we specify an error tolerance value for the right-hand side in Eq. (11) to be 180, so that when the $L_\infty$ norm value is less than 10, the algorithm will stop. Using this, the 35$^{th}$ aFVF is selected. For the 45-degree line correspondence stopping criterion described in Section 4, we set $\epsilon$ equal to 0, $\xi$ equal to 0.005 and $m$ equal to 5. The idea is if $\delta$ is less than or equal to 0, then the $b_1''$ and $b_1'$ curves have crossed, indicating the algorithm is nearing the point when it should stop. With this setting, the 43$^{rd}$ aFVF is selected since at the 43$^{rd}$ iteration, $\delta$ is equal to -0.003, and from 44$^{th}$ to 48$^{th}$ iteration, the $\delta$ values are between -0.001 and 0.002, which are all less than 0.005.

Next we simulate the two identified aFVFs to assess if the stopping rules yielded good ADP policies. For the simulation, 100 scenarios are conducted by initializing the state variables in the first stage using a Sobol sequence with the same range as shown in Table 1. The simulation is executed for 70 time periods for each initial point, following the same procedure as Chen et al. (2017). In addition, the solution policy from the greedy algorithm is used as a benchmark. The main difference between the greedy algorithm and ADP is that the greedy algorithm does not consider the future state. The details of the greedy algorithm used can be found in Chen et al. (2017).

After simulating these two ADP policies and the greedy policy, the mean costs of the 100 scenarios of these three policies are shown in Table 2. 2.

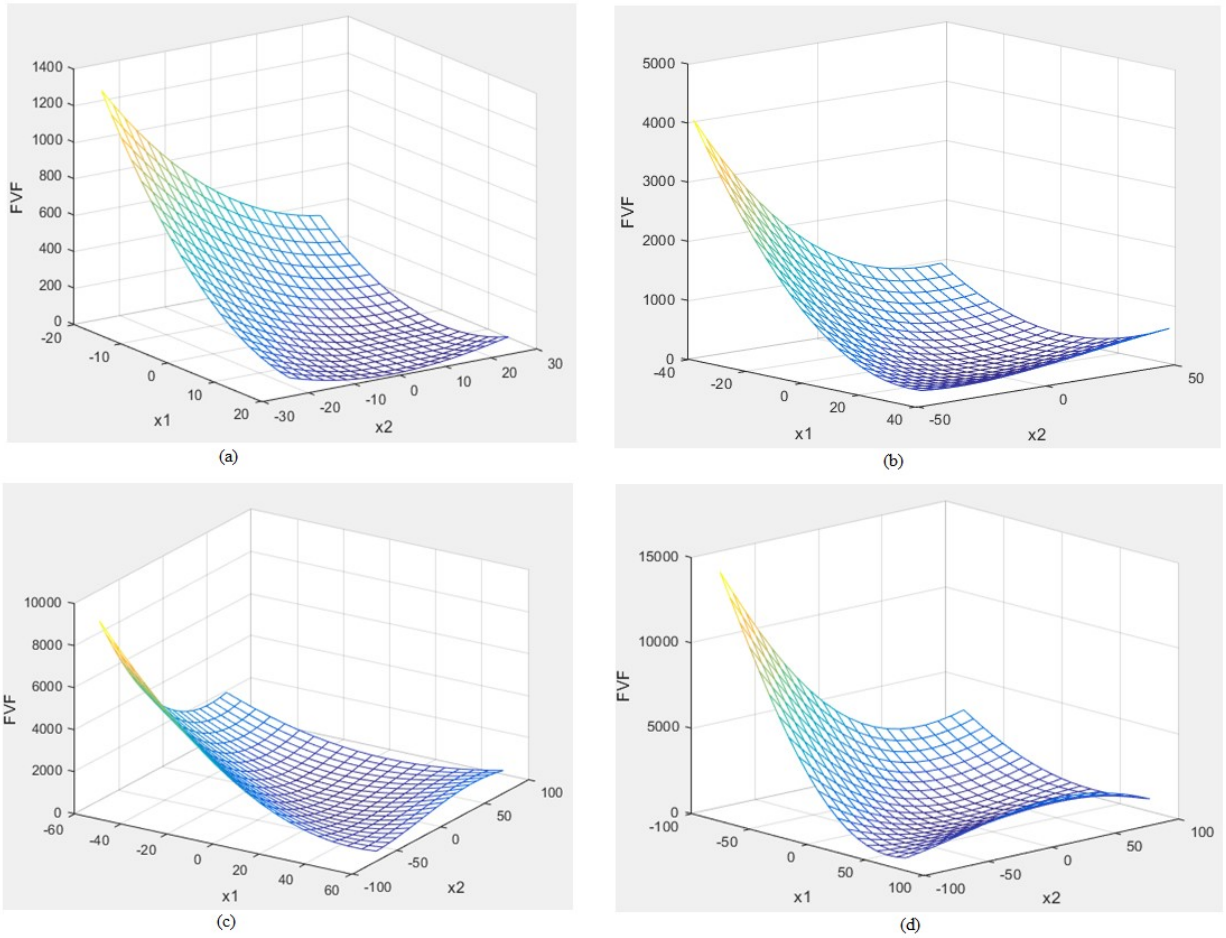
Table 2. 2 Mean cost from simulating the 100 scenarios for the three policies.

| Policy | 35th aFVF | 43rd aFVF | Greedy policy |
|---|---|---|---|
| Mean cost ($) | 294.98 | 295.52 | 331.54 |

From this table, it can be seen that the ADP policies achieve much lower mean costs than the greedy policy. The mean cost of the 35th aFVF, which was selected by the $L_\infty$ norm rule, is only slightly lower than the one for the 43rd aFVF, which was selected by the 45-degree line correspondence rule. In order to distinguish the difference between these two ADP policies, we conduct a paired *t*-test on the simulation results as conducted in Chen et al. (2017). Comparing the simulation outputs of the 100 scenarios from 35th aFVF and 43rd aFVF, the *t*-test *p*-value is 0.031. This indicates statistically that the 43rd aFVF is only marginally better than the 35th aFVF. By contrast, a paired *t*-test between the 43rd aFVF and greedy policy yields a *p*-value of 0.001, which indicates this ADP policy is statistically better than the greedy policy.

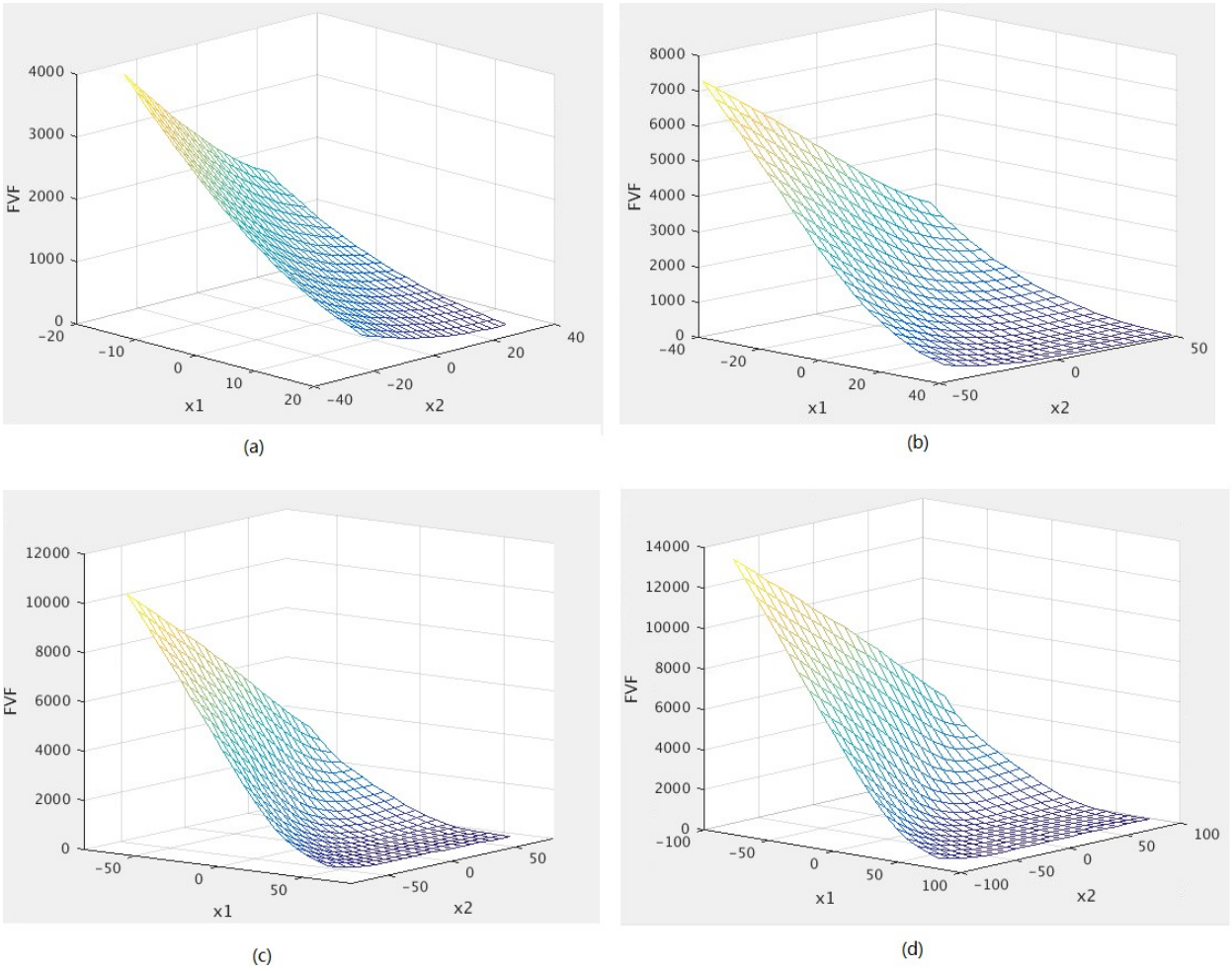
### 5.2. Extrapolation Investigation using SVR vs. MARS

Even though SVR has had remarkable success in machine learning, there has been little investigation of the impact of extrapolating an SVR model outside the training data region. For ADP, Lee and Lee (2004) suggested that the value function should be limited within the given state space region due to the uncertainty with extrapolation. However, when simulating a stochastic system, sometimes the system transitions to states that are beyond the given state space region. In this situation, decisions may be inaccurate. Hence, concerning this issue, we inspect how aFVFs created by SVR perform with occasional extrapolation.




Chapter 2. Figure 6. 3D meshplot of SVR 43rd aFVF with different plot ranges: (a) plot with the original range; (b) plot with double the original range; (c) plot with triple the original range; (d) plot with quadruple the original range.

First, to graphically illustrate the extrapolation issue, consider the 3D meshplots in Fig. 6. In this figure, we use the same aFVF, but change the plot scale to generate four meshplots. In this figure, x1 indicates inventory level of product 1 and x2 denotes the inventory level of product 2. From these four meshplots, it is clear to observe that when increasing the original range to the quadrupled level, the proper convex shape (Chen et al. 1999) of the FVF is lost, which indicates that incorrect decisions might be made when the values of state variables fall far enough outside of the original range.




Chapter 2. Figure 7. 3D meshplot of MARS high-quality aFVF with same ranges as Fig. 6

In order to conduct a comparison with MARS, Fig. 7 generates the corresponding four meshplots using the 11$^{th}$ aFVF identified by Chen et al. (2017). Compared to Fig. 6, the shape of the FVF is a proper convex shape in all four meshplots, which indicates that MARS is less susceptible to extrapolation issues than SVR. In Fig. 8, boxplots are shown of the simulation results for the 100 scenarios using the11$^{th}$ aFVF by MARS and the 43$^{rd}$ aFVF by SVR. The boxplots are similar, with the MARS policy showing a slightly lower median (the horizontal line inside the box), and the SVR policy showing a slightly smaller spread (the length of the box). The mean costs of MARS policy and SVR policy in the simulation are 296.67 and 295.52, respectively. A paired *t*-test on these two policies yields a *p*-value of 0.489, which indicates no statistical difference between these two policies. It is noted that when approximating the value

function, SVR and MARS both will result in approximation error, but from this result, it seems the extrapolation error caused by SVR is not significant. However, in the next subsection, we will further explore the extrapolation issue for SVR.

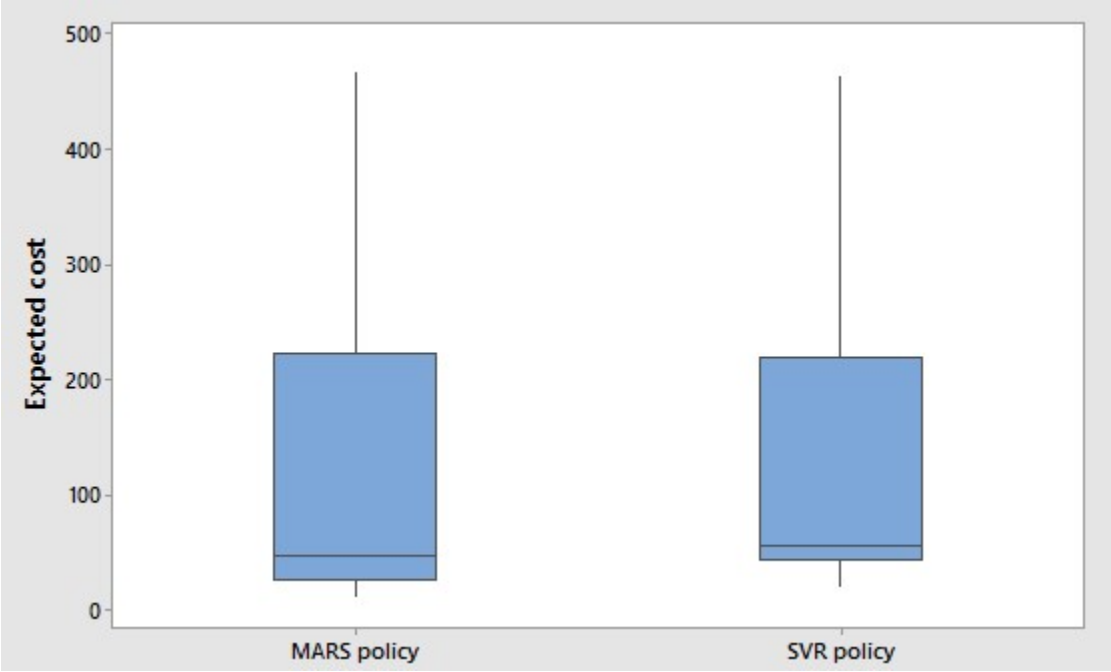



Figure 8

Chapter 2. Figure 8. Boxplot of expected total cost between MARS policy and SVR policy

### 5.3. Closer Investigation of Extrapolation for SVR

For each state vector, there are nine variables. Since we conduct 100 scenarios and each evolves over 70 periods, there are a total of 63,000 opportunities for extrapolation in any state dimension. Table 3 shows the number of extrapolation events for each state dimension. The total percentage of extrapolation events is 1.2%. Furthermore, after observing the value of these extrapolated components, we find that most components are within triple the original range. The exception is some cases for x2, for which there are 24 events that are less than -72, which is triple its lower limit. In order to investigate how significant the extrapolation error is, a further study is conducted below.

Table 2. 3 Number of extrapolation of each component of the state

| variables | x1 | x2 | x3 | x4 | x5 | x6 | x7 | x8 | x9 |
|---|---|---|---|---|---|---|---|---|---|
| Number of extrapolation | 10 | 297 | 143 | 7 | 14 | 5 | 126 | 141 | 69 |

Table 2. 4 Enlarged range of each state variable for the inventory stochastic DP problem

| # of Variable | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 |
|---|---|---|---|---|---|---|---|---|---|
| Min | -40 | -48 | -30 | 0 | 0 | 0 | 0 | 0 | 0 |
| Max | 40 | 48 | 30 | 40 | 48 | 30 | 26 | 32 | 20 |

As shown in Fig. 6, in the plot that doubles the original range, the convex shape of FVF is still observed. This indicates that SVR with the Gaussian RBF kernel still can perform well with limited extrapolation. Therefore, if we keep the initial range of state variable in the simulation unchanged, but enlarge the state space range when building the aFVFs, we may overcome the extrapolation error when simulating. On the basis of this idea, we enlarged the original state space range when building the aFVFs, as shown in Table 2. 4. Using the 45-degree line correspondence stopping rule as in Section 5.1, the $30^{th}$ aFVF is selected. The simulated mean cost for the $30^{th}$ aFVF policy built using the enlarged range is 297.29, compared to 295.52 for the $43^{rd}$ aFVF policy built using the original range. A paired *t*-test yields a *p*-value of 0.501, which indicates no statistical difference. Hence, this comparison indicates that the extrapolation error for SVR is not a significant issue. However, it should be noted that for the enlarged state space, the statistical modeling problem is more challenging and could require more training data to yield an accurate aFVF. Specifically, in this case, this $30^{th}$ aFVF required 600 training data points, but the $43^{rd}$ aFVF built using the original range, only required 150 training data points.

## 6. Concluding Remarks

In this paper, we employed SVR with the RBF kernel to implement the DACE based infinite horizon ADP algorithm introduced by Chen et al. (2017). Comparisons with the version using MARS illustrate improved behavior of the algorithm using SVR. Further, for the SVR version of the algorithm we specified and tested a stopping rule using the 45-degree line correspondence criterion proposed by Chen et al. (2017). ADP policies using this stopping rule and the $L_\infty$ norm stopping rule are compared and seem to be statistically similar using a paired *t*-test. In a study of extrapolation of the aFVF, a potential disadvantage of SVR vs. MARS is identified in which SVR

exhibits visually undesirable (nonconvex) behavior, while MARS seemingly maintains convex behavior. However, the impact of this potential extrapolation error with SVR is demonstrated to be minimal, and good ADP policies trained with different state ranges using SVR, are seen to be statistically similar through a paired *t*-test. Overall, while both SVR and MARS can yield good ADP policies for high-dimensional continuous-state stochastic DP problems, the SVR implementation demonstrates behavior over the DP loop iterations that is easier to control with formal stopping rules. However, the recommended parameter settings are intuitive and are anticipated to work well in general. Further testing can be conducted on parameters for stopping the algorithm using the 45-degree line correspondence rule.

**Acknowledge**

This research is partly supported by National Science Foundation grant ECCS-1128871.

**Reference**

Anderson, R. N., Powell, W. B., Scott, W. (2011). Adaptive Stochastic Control for the Smart Grid. Proceeding of IEEE 99(6): 1098-1115.

Bellman, R.E. (1957). Dynamic Programming. Princeton, NJ: Princeton University Press.

Bi, J., Bennett, K., Embrechts, M., Breneman, C. M., Song, M. (2003) Dimensionality reduction via sparse support vector machines. Journal of Machine Learning Research, 3: 1229-1243.

Bertsekas, D. P. (2017). Dynamic Programming and Optimal Control. Vol. I, 4th Ed. Athena Scientific.

Bertsekas D. P., Tsitsiklis J. N. (1996) Neuro-dynamic programming, Athena Scientific.

Castelletti, A., de Rigo, D., Rizzoli, A. E., Soncini-Sessa, R.,Weber, E. (2007). Neuro-dynamic programming for designing water reservoir network management policies. Control Engineering Practice 15, pp 1031-1038.

Castelletti, A., Galelli, S., Restelli, M., Soncini-Sessa, R. (2010). Tree-based reinforcement learning for optimal water reservoir operation, Water Resources Research. Vol. 46, W09507.

Cervellera, C., Wen, A., Chen, V. C. P. (2007). "Neural Network and Regression Spline Value Function Approximations for Stochastic Dynamic Programming." Computers and Operations Research, 34(1), pp. 70–90.

Cervellera, C. and D. Macciò (2011). A comparison of global and semi-local approximation in T-stage stochastic optimization. European Journal of Operational Research, 208, pp. 109-118.

Cervellera, C. and D. Macciò (2016). F-Discrepancy for Efficient Sampling in Approximate Dynamic Programming. IEEE Transactions on Cybernetics, 46(7), pp. 1628-1639.

Chen K., Wang, C. (2007). A hybrid SARIMA and support vector machines in forecasting the production values of the machinery industry in Taiwan. Expert Systems with Applications. 32(1): 254-264.

Chen, V. C. P., Ruppert, D., Shoemaker, C. A. (1999). Applying Experimental Design and Regression Splines to High-Dimensional Continuous-State Stochastic Dynamic Programming. Operations Research, 47, pp. 38–53.

Chen, V. C. P., Tsui, K. L., Barton, R. R., Meckesheimer, M. (2006). A review on design, modeling and applications of computer experiments. IIE Transactions. 38(4): 273-291.

Chen, Y., Liu, F., Kulvanitchaiyanunt, A., Chen, V. C. P., Rosenberger, J. (2017). Infinite Horizon Approximate Dynamic Programming Using Computer Experiments. COSMOS 17-02, University of Texas at Arlington.

Cortes, C., Vapnik, V. (1995). Support-Vector Networks. Machine Learning, 20, 273-297.

De Brabanter, K., Karsmakers, P., Ojeda, F., Alzate, C., De Brabanter, J., Pelckmans, K., De Moor, B., Vandewalle, J., Suykens, J. A. K. (2011). LS-SVMlab Toolbox User's Guide version 1.8. ESAT-SISTA Technical Report 10-146, August.

Drucker, H., Burges, C. J. C., Kaufman, L., Smola, A. J., Vapnik, V. N. (1997). Support Vector Regression Machines, Advances in Neural Information Processing Systems 9, NIPS 1996, 155–161, MIT Press.

Dong, B., Cao, C., Lee, S. E. (2005). Applying support vector machines to predict building energy consumption in tropical region. Energy and Buildings, 37(5): 545-553.

Farquad, M. A. H., Ravi, V., Bapi Raju, S. (2012). Analytical CRM in banking and finance using SVM: a modified active learning-based rule extraction approach. International Journal of Electronic Customer Relationship Management. 6(1): 48-73.

Fan, H., Tarun, P. K., Chen V. C. P. (2013). Adaptive Value Function Approximation for Continuous-State Stochastic Dynamic Programming. Computers and Operations Research, 40, pp. 1076–1084.

Furey, T. S., Cristianini, N., Duffy, N., Bednarski, D. W., Schummer, M., Haussler, D. (2000). Support vector machine classification and validation of cancer tissue samples using microarray expression data. Bioinformatics. 16(10): 906-914.

Heath, D. C., Jackson, P. L. (1994). Modelling the evolution of demand forecasts with application to safety stock analysis in production/distribution systems. IIE Transactions. 26(3): 17-30.

Halton, J. H. (1960). On the efficiency of certain quasi-random sequences of points in evaluating multi-dimensional integrals. Numerische Mathematik, 2, pp. 84-90.

Hua, S., Sun, Z. (2001). Support vector machine approach for protein subcellular localization prediction. Bioinformatics. 17(8): 721-728.

Lee, J., Lee, J. H. (2004). Approximate Dynamic Programming Strategies and Their Applicability for Process Control: A Review and Future Directions. International Journal of Control, Automation, and Systems, vol. 2, no. 3, pp. 263-278.

Li, T. Huang, C. (2009). Defect spatial pattern recognition using a hybrid SOM-SVM approach in semiconductor manufacturing. Expert Systems with Applications, 36(1): 374-385.

Martinize-de-Pison, F. J., Barreto, C., Pernia, A., Alba, F. (2008). Modelling of an elastomer profile extrusion process using support vector machines (SVM). Journal of Materials Processing Technology. 197(1-3): 161-169.

Mohandes, M. A., Halawani, T. O., Rehman, S., Hussain, A. A. (2004). Support vector machines for wind speed prediction. Renewable Energy. 29(6): 939-947.

Powell, W. B. (2007). Approximate dynamic programming: solving the curses of dimensionality. Wiley, New York.

Rick, C., Zhong, W., Blackmon, M., Stolz, R., Dowell, M. (2008). An efficient SVM-GA feature selection model for large healthcare databases. Proceedings of the 10th annual conference on Genetic and evolutionary, pp. 1373-1380, Atlanta, GA, USA.

Sarikprueck, P., Lee, W., Kulvanitchaiyanunt, A., Chen, V. C. P., Rosenberger, J. M., (2015). Novel Hybrid Market Price Forecasting Method With Data Clustering Techniques for EV Charging Station Application. IEEE Transactions on Industry Applications. 51(3):1987-1996.

Simao, H. P., Day, J., George, A. P., Gifford, T., Nienow, J., Powell, W. B. (2008). An Approximate Dynamic Programming Algorithm for Large-Scale Fleet Management: A Case Application. Transportation Science. 43(2): 178-197.

Sobol, I. M. (1967). The distribution of points in a cube and the approximate evaluation of integrals. USSR Computational Mathematics and Mathematical Physics, 7, pp. 784-802.

Suykens, J. A. K., Van Gestel, T., De Brahanter, J., De Moor, B., Vandewalle, J. (2002) Least squares support vector machines. World Scientific Pub. Co. Singapore.

Sutton, R. S., Barto, A. (1998) Reinforcement learning: an introduction. *The MIT Press*, Cambridge, Massachusetts.

Van Roy, B., Bertsekas, D., Lee, Y. (1997). A Neuro-Dynamic Programming Approach to Retailer Inventory Management. Proceedings of the 36th IEEE Conference on Decision and Control, 12-12 Dec.

Wei, Q., Liu, D., Shi, G. (2015). A Novel Dual Iterative Q-Learning Method for Optimal Battery Management in Smart Residential Environments. IEEE Transactions on Industrial Electronics. 64(4): 2509-2518.

Yang, Z., Chen, V. C. P., Chang, M. E., Sattler, M. L., Wen, A. (2009). A decision-making framework for ozone pollution control. Operations Research, 57(2), pp. 484–498.

Yao, J., Lian, C. (2016). A New Ensemble Model based Support Vector Machine for Credit Assessing. International Journal of Grid and Distributed Computing, 9(6): 159-168.

Zhang, L., Hu, H., Zhang, D. (2015). A credit risk assessment model based on SVM for small and medium enterprises in supply chain finance. Financial Innovation, (2015) 1:14.

Zhou, W., Zhang, L., Jiao, L. (2002). Linear programming support vector machines. Pattern Recognition, 35: 2927-2936.

## Chapter 3. Approximate Dynamic Programming for Control of a System of Electric Vehicle Charging Stations

Abstract

Taking the Dallas Fort-Worth metropolitan area as the assumed sample system, control of an electric vehicle (EV) quick charging infrastructure with 11 EV charging stations is performed. This system integrates renewable energy to supply electricity to charging stations as well as the main electrical grid. Batteries as electrical storage facilities are installed in each station. Through the usage of batteries, the control system is expected to be able to store the surplus electricity from renewable energy or buy electricity at a lower electricity market price (EMP) and sell the stored electricity back to the main grid when the EMP reaches to a higher level to decrease the operational cost or make profits for this system. In order to control this system, we formulate it as a Markov decision process problem. An infinite horizon approximate dynamic programming approach based on design and analysis of computer experiments concept is used to solve this high-dimensional, large-scale, EV charging station control problem over continuous spaces. A 45-degree line correspondence stopping criterion specified is used to stop the DP iterations early and select the ADP policy. From the results, it is clear that control from the selected ADP policy has a better performance than the benchmark policy. In addition, a random selected ADP policy from later iteration has the same performance as the one selected by the stopping rule statistically, which indicates the quantified stopping rule is able to stop DP iterations reasonably and reduce a lot of computational time.

Key words: electric vehicle charging infrastructure, approximate dynamic programming, renewable energy, design and analysis of computer experiments, large-scale

### 1. Introduction

Environmental pollution is becoming more and more serious with the rapid development of our society (Chen et al. 2013). Traditional energy sources such as coal, gas, and oil are some of the main sources resulting in pollution such as greenhouse effects, and dust and ashes in the air. Consequently, research on renewable energy sources is becoming more prevalent. With developing technology, greenhouse gas emissions will be reduced to 17% of 2005 levels by 2020 as the U.S. government pledged (http://www.eia.doe.gov/emeu/aer/pdf/pages/sec12_4.pdf, 2010).

After investigation and analysis, the transportation sector, causing up to 33.1% of all energy related emissions, has been identified as the largest producer of carbon dioxide emission in the U.S. (http://cta.ornl.gov/data/download29.shtml). Therefore, to abate such emissions, electric vehicles (EVs) have been given a lot of attention. This fundamental transformation from oil based vehicles to electric power ones will help decrease carbon dioxide emissions substantially. From Fell et al. (2010), there will be more than 1 million EVs in US by 2017 with the target growth rate and in the Dallas-Fort Worth (DFW) metro area, the amount of EVs will be increased to around 10,000 by that time.

In order to increase the penetration and usage of EVs, the drivers' anxiety about the driving range should be resolved first. In existing literature, the design of optimal EV charging profiles is a way to serve this purpose. For example, the authors take advantage of charging behaviors from demographical statistical data to assess the EV charging scenarios (Steen et al. 2012), vehicle usage data to predict and analyze the EV charging profile (Ashtari et al. 2012) and a dynamic game theoretic optimization to formulate the optimal EV charging problem (Zhu et a. 2012). Additionally, in Clement-Nyns et al. (2010), coordinated charging profile with the objective of minimizing power losses and maximizing the main grid load factor in a residential distribution, where the optimal EV charging stations are included, is presented. Yao et al. (2017), considering demand response, formulated the optimal power profile with the objective of maximizing the number of charging EVs and minimizing the total charging cost simultaneously. Furthermore, other models for coordinated EV charging systems have been developed to improve power utilization (Wang et al. 2015), smooth real-time power fluctuation in a regulation service (Shaaban et al. 2014), avoid overload in the utility grid (Gan et al. 2013), and support the power system restoration as a black-start power source (Sun et al. 2016). In order to improve the penetration of sustainable energy in the charging systems, Badawy and Sozer (2017), Khodayar et al. (2012), Marano and Rizzoni (2008), Guo et al. (2014), etc, have employed wind or solar generation as a type of energy resources to supply electricity to the charging stations.

Though these studies have proposed various different effective methods to develop EV charging infrastructure, they focus on level 1 or level 2 charging systems, which still cannot meet the desire of fast charging from the customers. Considering this issue, Sarikprueck et al. (2017) proposed a novel regional EV DC level 3 fast charging system equipped with renewable resources such as wind and solar energy, to serve EV demand within minutes. However, the

decision making procedure in Sarikprueck et al. (2017) is deterministic and when making decisions, the system does not consider the future state but only the current state. Kulvanitchaiyanunt et al. (2016) utilized this system, which makes a decision for each stage through linear programming without considering the uncertainty.

Therefore, for a public charging station development, we also focus on this EV DC level 3 fast charging system which was initially well designed in Sarikprueck et al. (2017) However, we formulate it as a Markov decision process (MDP) problem so that the future state needs to be considered when making a decision. In this system, wind and solar energy provide electricity as well as main power grid. Besides, in each station, local battery storage systems are used as a buffer to minimize the operational cost. However, different from Kulvanitchaiyanunt et al. (2016) and Sarikprueck et al. (2017), we apply approximate dynamic programming (ADP) to solving this large-scale, high-dimensional, dynamic control system so that the decision making procedure considers the uncertainty.

In the rest of this paper, background and contribution are introduced in section II. Then, the details of this dynamic EV charging station control problem are introduced in section III. The computational results using SDP algorithm are shown in section IV. Moreover, discussion about the simulation results is proposed in section V. Finally, a conclusion about this research is presented.

## 2. Background

In the literature, ADP algorithms such as Q-learning and post-decision state approach, have been used a lot in the smart grid related research as a control technique since such system evolves over time and the problem in it is a multi-stage optimization problem.

Namely, Wei et al. (2015) proposed a novel iterative Q-learning method named “dual iterative Q-learning algorithm” to solve the optimal battery management control problem in smart residential environments. Based on this research, later on, Wei et al. (2017) introduced a mixed iterative adaptive DP approach to control the battery energy in smart residential micro-grids. On the basis of the idea of smart grid, Boaro et al. (2013) presented an intelligent management scheme for renewable energy combined with battery implemented with a faster and simpler

scheme of dynamic programming. Through online learning control system which is based on the fundamental principle of reinforcement learning (RL) or more specifically neural DP, the optimization of electricity consumption in office buildings was conducted in Shi et al. (2016). In addition, reactive power control of grid-connected wind farm was solved by an adaptive DP in Tang et al. (2014). Moreover, a heuristic DP (HDP) architecture was established to schedule the quality of service in cognitive-ratio-based smart grid networks in Yu et al. (2016). Furthermore, a modified ADP based on actor-critic network in Xie et al. (2016) was used to schedule fair energy to vehicle-to-grid networks. Besides, Jiang and Powell (2015) employed a convergent ADP algorithm that exploits monotonicity of the value function to find a revenue-generating bidding policy in the real-time electricity market with battery storage. According to these latest researches, it is clear to find out that ADP algorithm is playing a critical role in the control of smart grid problems.

However, as to the large-scale, high-dimensional DP problems over continuous spaces, it is still a challenge for the algorithms used in the above studies. For example, although, in Yu et al. (2016), there were 28 state variables and 8 decision variables in the space, these variables were discrete as the same as in Shi et al. (2016) and Xie et al. (2016). If considering continuous spaces, the systems are not very large-scale. For example, Ernst et al. (2009) considered only two state variables and one decision variable, and Wei et al. (2017) only used two state variables and two decisions variables over continuous spaces. Even though a deep learning based RL algorithm is developed in Peng et al. (2016) aiming to solve the high-dimensional infinite horizon DP problem over a continuous state space, this methodology requires a large amount of data, including 300,000 iterations of training and 10 million tuples, which denotes the algorithm requires substantial computation. In addition, they optimized over a discrete action space using a derivative-free evolutionary algorithm. As Lee and Lee [28] summarized, the common limitations of using RL and neuro-dynamic programming (NDP) are: a) in DP problems with continuous state and action spaces, the discretization and common "incremental" update rule are not practical, and the function approximation errors can grow rapidly; b) complex dynamics of most chemical processes still bound the state space that can be explored, which results in regions of sparse data.

To solve the large-scale, high-dimensional, infinite horizon DP problem over continuous spaces, Chen et al. (2017a) proposed design and analysis of computer experiments (DACE) based infinite horizon ADP algorithm to sample the state space with design of experiment and

approximate the value function via statistical modeling methods. This algorithm is based on the finite horizon version proposed by Chen et al. (1999). With the approach introduced in Chen et al. (1999), several large-scale, high-dimensional, finite horizon DP cases have been solved successfully such as more than 500 dimensions ozone problem (Yang et al. 2009) and 38 dimensions waste water problem (Cervellera et al. 2006). Hence, considering the limitations in RL/NDP and success achieved by DACE based finite horizon algorithm, we employ this DACE based infinite horizon ADP algorithm introduced in Chen et al. (2017a) to solve this large-scale, high-dimensional, infinite horizon, EV charging station control problem over continuous spaces. Therefore, the contribution of this paper is listed below:

(1) This EV Level 3 DC charging station system is formulated as a MDP problem.
(2) State transition models are developed based on support vector regression (SVR) and martingale model of forecast evolution (MMFE) models.
(3) A large-scale, high-dimensional, infinite horizon, DP problem over a continuous space is solved with an ADP algorithm

3. ADP approach

In this section, first, we will briefly describe the infinite horizon stochastic DP (SDP) algorithm and then overview the DACE based infinite horizon ADP algorithm that will be used in this study.

### 3.1. Infinite horizon SDP model

DP as a mathematical programming approach to optimize a system evolving over time was introduced by Bellman in 1957. There are two versions of DP: finite horizon and infinite horizon. Different from finite horizon DP, infinite horizon DP only has one true value function. A typical recursive SDP formulation for an infinite horizon problem can be written as

$$V(x) = \min_{u \in \Gamma} E\{c(x,u,\zeta) + \gamma V(f(x,u,\zeta))\} \quad (1)$$

In Eq. (1), $x$ is vector of state variables, $u$ is a vector of decision variables, $\zeta$ is a vector of stochastic variables, $f$ is the transition function, $\Gamma$ is a set of feasible decisions, $\gamma$ is a discount factor, $c$ is the cost function, and $V$ is the future value function (FVF). As described in Section **Error! Reference source not found.**, finding an FVF exactly is intractable for medium-sized

problems. Consequently, ADP attempts to find a converged approximate FVF (aFVF) ($\hat{V}$) using the following formulation.

$$\hat{V}(x) \approx \tilde{V}(x) = \min_{u \in \Gamma} E\{c(x,u,\zeta) + \gamma \hat{V}(f(x,u,\zeta))\} \quad (2)$$

### 3.2. Overview of DACE based infinite horizon ADP algorithm

In order to solve large-scale infinite horizon DP problems over continuous spaces, Chen et al. (2017a) proposed an algorithm as shown below, on the basis of DACE concept and adaptive value function approximation (AVFA) approach (Fan et al. 2013).

Step 0: Initialization:

(a). Input a discount factor $\gamma$, a state transition function $f$, and a cost function, $c$.

(b). Choose the training data set $X^{\text{Train}}$ and testing data $X^{\text{Test}}$ set generated by low-discrepancy sequences, respectively.

(c). Set the iteration counter to $k \leftarrow 0$, set the initial aFVF $\hat{V}_0 = 0$, and the set of evaluated state variables $X \leftarrow \emptyset$.

Step 1: Iteration of infinite horizon DP:

(a). Set $k \leftarrow k + 1$.

(b). For each state variable $x \in X^{\text{Train}} \cup X^{\text{Test}} - X$, solve $\tilde{V}_k(x) = \min_{u \in \Gamma} E\{c(x,u,\xi) + \gamma \hat{V}_{k-1}(f(x,u,\xi))\}$ and set $X \leftarrow X \cup \{x\}$;

(c). Fit a regression model using the data $\{(x, \tilde{V}_k(x))\}_{x \in X^{\text{Train}}}$ to obtain $\hat{V}_k$;

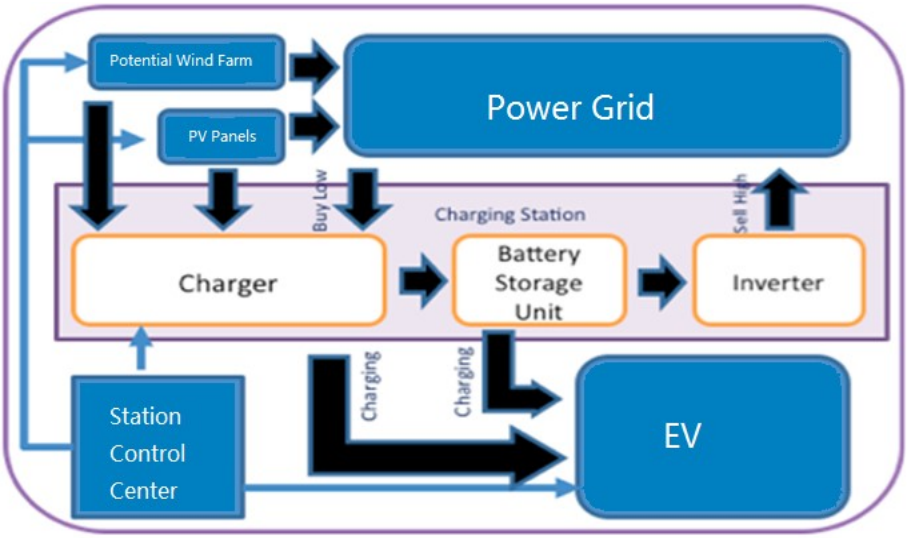
(d). If $\hat{V}_k$ fails the stopping criteria for the data loop on the data $\{(x, \hat{V}_k(x))\}_{x \in X^{\text{Test}}}$, a new set of state variables $X'$ will be added to the training data set $X^{\text{Train}} \leftarrow X^{\text{Train}} \cup X'$ and go to the step 1 (b);

In this algorithm, two loops are used to achieve the aFVF: an inner data loop and an outer DP loop. The data loop follows the AVFA approach of Fan et al. (2013) to sample the state space sequentially in order to control the amount of sampling needed to build an aFVF. The DP loop follows the value iteration concept to generate a sequence of aFVFs. In the above description, Steps 1(a)-(e) are the DP loop and Steps 1(b)-(d) represent the data loop. As presented in Chen et al. (2017b), SVR has presented a more stable performance in the value function approximation

than multivariate adaptive regression splines for the infinite horizon DP case. Therefore, in this study, SVR with Gaussian radial basis function (RBF) kernel is also used to approximate the FVF. As the same as Chen et al. (2017b), in this study, we continue to use least-square support vector machines toolbox downloaded from http://www.esat.kuleuven.be/sista/lssvmlab/ to create the aFVFs. As to the tuning process for the parameters of RBF kernel, please refer to Brabanter et al. (2011) for details.

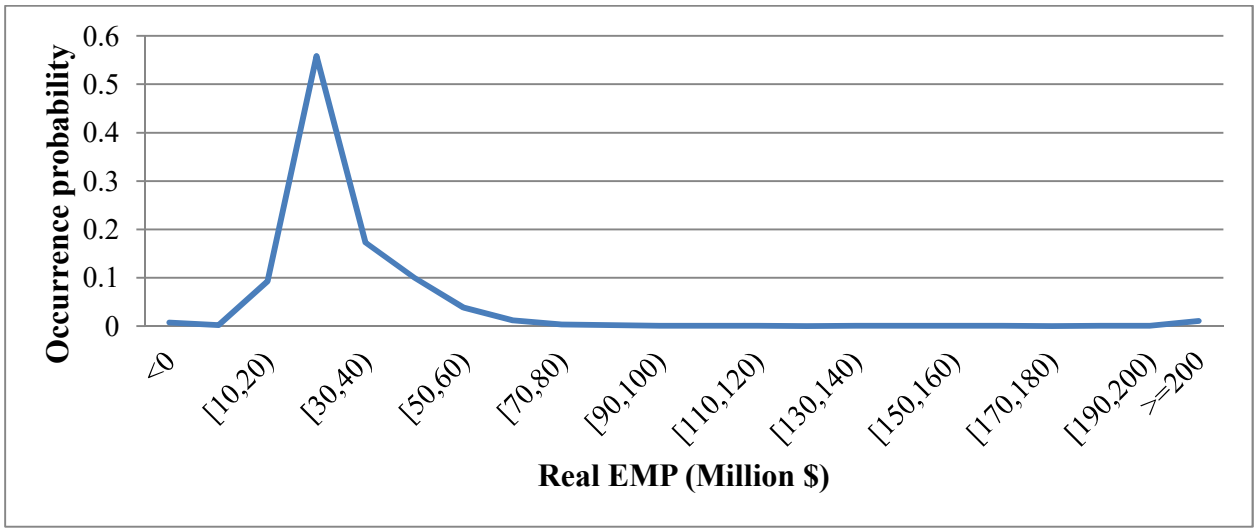
## 4. Dynamic control problem formulation

In this EV charging station control problem, following the design in Sarikprueck et al. (2017), it is assumed that there are 11 stations located at DFW metropolitan area. In this system, a remote wind farm provides wind power for the system through a long term contract and solar panels installed on the roof of each station supply the solar power for each station. Main grid also exchanges the electricity to each station. A battery is installed in each station for electricity storage and the priority of this control is to satisfy the demand from EVs and simultaneously, make optimal decisions at each time period for electricity trading with the main grid to minimize the operational cost for the system. The blueprint of this dynamic control system is shown blow. Ideally, the control strategy is that when the electricity market price (EMP) is low, the system will buy electricity and store the surplus one in the battery, and sell electricity when the EMP is high. In this study, based on the technology of photovoltaic panels, we assume solar power only provides a very small amount of electricity to the system and in addition, the maximum supply of wind power is no more than 30% of total demand from EVs. Therefore, EMP plays a critical role in this power trading system.




Chapter 3. Figure 1 Blueprint of this EV charging station control system (Sarikprueck et al. 2017)

According to the historical operation record, EMP has two types: spike ones and non-spike ones and the historical EMP range is from -150 MW to 3300 MW. However, the real EMP is distributed as a Cauthy distribution and the occurrence of the negative and spike EMPs are rare as shown below.




Chapter 3. Figure 2 Real EMP distribution

In statistics, when building the model with such distribution, the accuracy of the model will be affected a lot by the rare data with large values (Kutner et al. 2004). Hence, based on the control strategy mentioned above, it is straightforward to control the system to sell electricity back to the main grid when positive spike EMP occurs and purchase electricity for making profits when the negative EMP occurs, which is defined as the first-type control strategy in this paper. The ADP policy is used to the other EMPs located in [0, 200), which is called as the second-type control strategy. In this paper, we mainly focus on developing the second-type control strategy.

### 4.1. EV charging station control problem formulation

Kulvanitchaiyanunt et al. (2016) formulate this EV control problem as a deterministic linear programming problem. Based on the formulation in Kulvanitchaiyanunt et al. (2016), we formulate it as a MDP problem as shown below.

Table 3. 1 Major nomenclature used

| | | | |
|---|---|---|---|
| $T$ | Total time period | $I$ | Inventory of battery |
| $D$ | Total demand | $D2$ | Demand satisfied by the battery of a station |
| $e$ | Battery storage efficiency | $R$ | Electricity sold back to the grid from battery |
| $W$ | Wind energy purchased from wind farm | $\widetilde{W}$ | Fraction of wind energy allocated to each station |
| $n$ | Total number of open stations | $PV$ | Photovoltaic (solar) production for each station |
| $N$ | Number of recourse scenarios | $D1$ | Demand satisfied by direct charge of each station |
| $I_{min}$ | Minimum battery storage level of each station | $I_{max}$ | Maximum battery storage level of each station |
| $g^+$ | Electricity bought from grid | $g^-$ | Electricity sold back to grid from direct charge |
| $BC$ | Amount of battery charged electricity | $EMP$ | Electricity market price |
| $cr$ | Battery charge upper limit | $dc$ | Battery discharge rate |

$$\sum_{t=1}^{T}\gamma^{t}\left(\frac{1}{N}\left(\sum_{i=1}^{N}\sum_{j=1}^{n}\left(EMP_{t,i}^{j}g_{t,i}^{+,j}-EMP_{t,i}^{j}\left(g_{t,i}^{-,j}+R_{t}^{j}\right)\right)\right)\right)\ ,\qquad \forall j\in n,\forall j\in n,\forall i\in N \quad (3)$$

Eq. (3) is the objective function of this SDP problem and the first constraint set includes the battery level transition from period *t*-1 to period *t* for each open station *j*:

$$I_{t}^{j}=I_{t-1}^{j}+BC_{t}^{j}-\frac{R_{t}^{j}}{e}-\frac{D2_{t}^{j}}{e}\qquad \forall j\in n,\ \forall i\in N\ . \quad (4)$$

The storage efficiency is assumed to be 79.8%. The solar generation is the same for each station. Therefore, the amount of battery charged electricity is calculated with following equation:

$$BC_{t}^{j}=\widetilde{W}_{t}^{j}W_{t,i}+PV_{t,i}+g_{t,i}^{+,j}-g_{t,i}^{-,j}-D1_{t}^{j}\qquad \forall j\in n,\ \forall i\in N\ . \quad (5)$$

The total demand consists of the demand satisfied by direct charge and demand satisfied by the battery as shown in the following constraint:

$$D_{t}^{j}=D1_{t}^{j}+D2_{t}^{j}\qquad \forall j\in n\ . \quad (6)$$

The combination of electricity sold back to the grid from the battery and demand satisfied by the battery together is less than or equal to the discharge rate (*dc*) multiplied by the storage efficiency, as shown below:

$$R_{t}^{j}+D2_{t}^{j}\leq dc\times e\qquad \forall j\in n\ . \quad (7)$$

The battery charge must not be greater than its charge upper limit. and the battery level must be constrainted in between the minimum battery level and the battery capacity for each station, as shown below:

$$BC_{t}^{j}\leq cr\qquad \forall j\in n\ , \quad (8)$$

$$I_{min}\leq I_{t}^{j}\leq I_{max}\qquad \forall j\in n\ . \quad (9)$$

The sum of the fraction of the wind allocation should be equal to 1:

$$\sum_{j=1}^{n}\widetilde{W}_{t}^{j}=1\qquad \forall j\in n\ , \quad (10)$$

$$I_t^j, \widetilde{W}_t^j, g_{t,i}^{+,j}, g_{t,i}^{-,j}, BC_{tj}, R_t^j, D1_t^j, D2_t^j \geq 0 \quad \forall j \in n, \forall i \in N \ . \tag{11}$$

As assumed, there are two type decision variables: one is first-stage decision variables and the other is recourse decision variables. To solve this, stochastic programming is used and defined decision variables are shown in the following table. It is noted that all these decision variables are in the continuous space.

Table 3. 2 Decision variables

| | |
|---|---|
| First-stage decision variables | $R_t^j$, $\widetilde{W}_t^j$, $D1_t^j$, $D2_t^j$, $I_t^j$, $BC_t^j$ $\quad \forall j \in n$ |
| recourse decision variables | $g_{t,i}^{-,j}$, $g_{t,i}^{+,j}$ $\forall j \in n, \forall i \in N$ |

### 4.2. State transition model

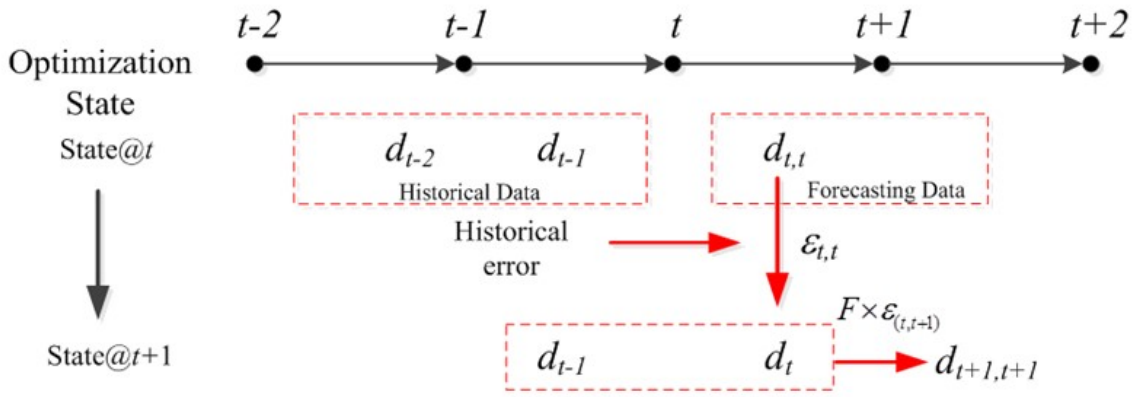
Using the features selected in Sarikprueck (2015) for wind power, solar power and non-spike EMP, through SVR with its parameter tuning process for RBF kernel, we build the forecasting models for wind power, solar power and non-spike positive EMP. In addition, Martingale model of forecast evolution (MMFE) (Heath and Jackson 1994) approach as an effective forecast uncertainty analysis method which is implemented to explore the stochastic nature of these forecasting models, is utilized in this research. Therefore, combining SVR forecasting models and MMFE, we have the following state transition process. In details, at the time period *t*, we have the historical states $d_{t-2}$, $d_{t-1}$ and forecasting state $d_{t,t}$; when evolving to time period $t + 1$ and keeping the same structure, the state vector becomes $d_{t-1}$, $d_t$ and $d_{t+1,t+1}$. From time period *t* to $t + 1$, $d_{t-2}$ changes to $d_{t-1}$, which is defined as identity transition used in Yang et al. (2009). And for the other state variables:

$$d_t = d_{t,t} \times \varepsilon_{t,t} \quad , \tag{12}$$

$$d_{t+1,t+1} = F(d_{t-1}, d_t) \times \varepsilon_{t,t+1} \quad , \tag{13}$$

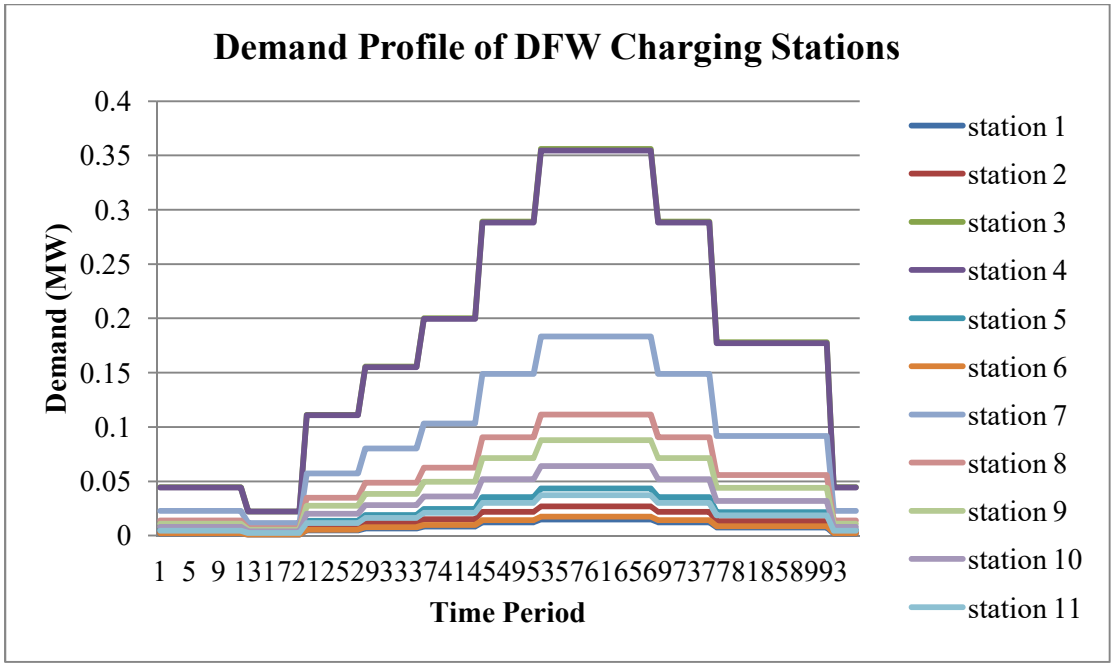
where $\varepsilon$ is generated using MMFE methods (please refer to Health and Jackson for details), $F$ is the forecasting models mentioned above, generated by SVR. Note that only wind power vector, solar power vector and EMP vector will use this transition process. In the state space, it also includes the battery inventory level and demand level. Battery inventory level will use the Eq. (4)

for the update. In this study, we make use of the DFW EV demand profile shown in Fig. 4, designed by Khosrojerdi et al. (2013). In this demand profile, Khosrojerdi et al. (2013), only several spikes occur, which indicates that demand only varies several times among 96 periods. Hence, when transitioning from time period $t$ to $t + 1$, we define the demand levels of stations are kept unchanged to approximate the FVFs. However, in the simulation, we take advantage of Fig. 4 for the evolving demand level.




Chapter 3. Figure 3 State Transition Process for wind, solar and EMP state variables

In wind power forecasting, wind speed is used as an important state variable and in the non-spike EMP forecasting, load profile and temperature are also included. As to these three state variables we take advantage of the historical related data to represent them at time period $t$ and shift them to the next period based on the historical data at time period $t + 1$. Therefore, when implementing DACE based infinite horizon ADP algorithm, in the state space, these three state variables are not included in the state space. Therefore, if there are 5 stations open in the state space, we have the following state variables: 3 for wind, 3 for solar variables, 15 for EMPs in five stations, 5 for battery inventory levels of five stations and 5 for the demand levels for each station. In total, the state space is 31 dimensions.




Chapter 3. Figure 4 Demand profile of 11 charging stations located in DFW metro area

Khosrojerdi et al. (2013)

## 5. Computational Results

In this section, first, we will overview the 45-degree line stopping criterion proposed in Chen et al. (2017a) and specified in (2017b), and then quantify the stopping settings in advance. After that, we will use ADP solution algorithm to build aFVFs iteratively until the stopping rule is satisfied. Moreover, the selected aFVF will be applied in the simulation in order to explore its performance. Greedy policy as a benchmark is used for comparison with ADP policy.

### 5.1. 45-degree line correspondence stopping criterion

45-degree line correspondence stopping criterion proposed in Chen et al. (2017a) is used to identify the shape of value function by observing the linear relationship between consecutive outputs of DP iterations. The fitted regression line is represented:

$$\hat{Y}_k = b_0 + b_1 X_{k-1} \quad , \tag{14}$$

where $b_0$ is the intercept, $b_1$ is the slope, $X$ is the $\tilde{V}$ at iteration $k$-1, $Y_k$ is the $\tilde{V}$ at iteration $k$, $\hat{Y}_k$ is the estimated $Y_k$.

For a better application, Chen et al. (2017b) presented an algorithm to specify this rule. In this algorithm, $b_1^{'}$, named as $b_1$with 1 lag, is the relationship between the consecutive outputs of $\tilde{V}$ at iteration $k$ and $k$ - 1. $b_1^{''}$, named as $b_1$ with 2 lags, is the relationship between the outputs of $\tilde{V}$ at iteration $k$ and $k$ - 2. This algorithm mainly takes advantage of the difference between $b_1^{'}$ and $b_1^{''}$ to identify the ADP policy and stop the DP iterations.

$$\delta_k = b_{1,k}^{''} - b_{1,k}^{'} \quad , \quad k \geq 3 \, , \qquad (15)$$

where $\delta_k$ is the difference between $b_1$ with 2 lags and $b_1$ with 1 lag at $k^{\text{th}}$ DP iteration.

According to this algorithm, if $\delta_k$ is less than or equal to $\epsilon$, which is a specified error tolerance value, $b_{1,k}^{''}$ and $b_{1,k}^{'}$will be close enough. At this time, a potential "good" ADP policy might have been generated and we need to check the $|\delta|$ values from iteration $k$ +1 to $k$ +$m$ to see if all $|\delta|$ values are also less than or equal to $\xi$. If so, output approximate value function ($\hat{V}_k$) at iteration $k$, otherwise, DP iterations should be continued. As to the detail of this algorithm, please refer Chen et al. (2017b).

### 5.2. FVF approximation

In this research, we randomly assume there are five stations open and the number of slots in these stations is different as shown below:

Table 3. 3 Operation condition of 11 charging stations in DFW metro area

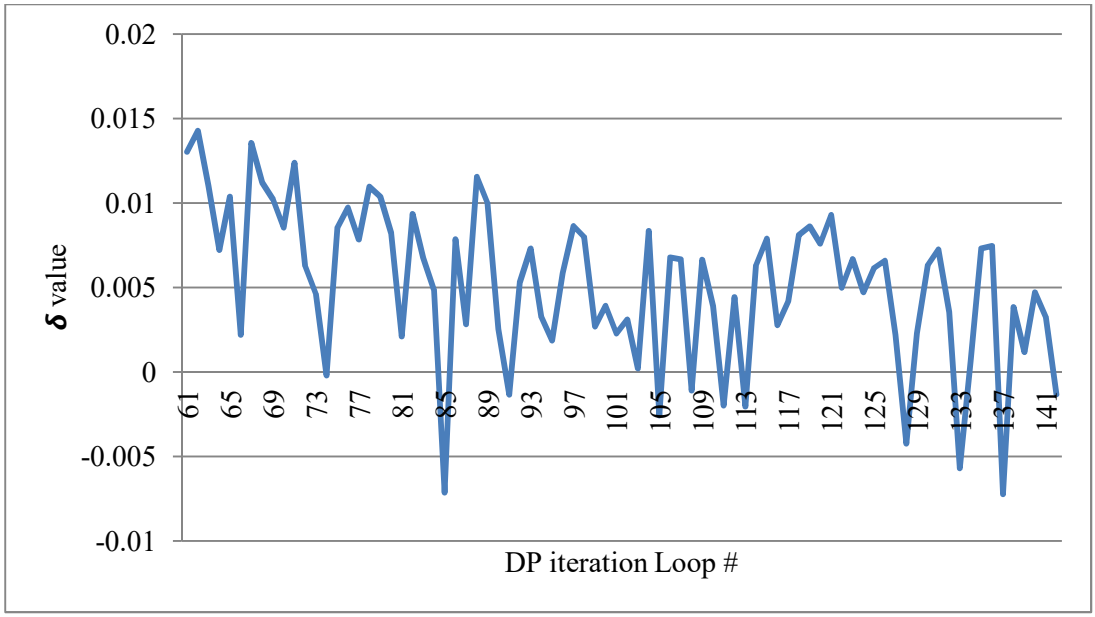
| Order of stations | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | 11 |
|---|---|---|---|---|---|---|---|---|---|---|---|
| # of slots | 0 | 2 | 9 | 0 | 6 | 0 | 4 | 0 | 0 | 8 | 0 |

Number of slots plays the key role in determining the size of battery in each station. In the following, it is the battery size for each open station:

Table 3. 4 Battery size of open stations

| Order of stations | 2 | 3 | 5 | 7 | 10 |
|---|---|---|---|---|---|
| Min level (MW) | 1.44 | 6.48 | 4.32 | 2.88 | 5.76 |
| Max level (MW) | 7.2 | 32.4 | 21.6 | 14.4 | 28.8 |

Since there are five stations open, the state space is 31 dimensions that are much larger than the related work in the literature. Following the algorithm proposed in Chen et al. (2017a), the size of training data set and testing data set is 250 and 250, respectively and the increment size of training data is 50. Sobol sequence (Sobol 1967) is used to generate the training data and Halton sequence (Halton 1960) is used to initialize the testing data. The discount factor is set to be 0.995 since the time period is 15 minutes. Experiments on the infinite horizon DACE based ADP algorithm are conducted in MATLAB 2016b on a Lenovo computer with a Xeon, 16-core, 2.8 GHz CPU. For the optimization process, "fmincon" from MATLAB is used. In order to compute $\tilde{V}$ of each sampled point in the state space, eight recourse scenarios based on eight realizations of stochastic variables are conducted. The data loop stopping criterion is when the difference of consecutive data loop $R^2$ from the testing data set is less than 0.01, the data loop stops and algorithm goes to next DP iteration, otherwise, more data will be added to the training data set for another loop. As to the DP loop stopping criterion, considering the convergence of value iteration based ADP algorithm, those two specified error tolerance value should be very small, therefore, we set the $\epsilon$ equal to 0, $\xi$ equal to 0.005 and $m$ equal to 5. This error tolerance setting indicates the high-quality ADP policy is selected at the point where the $b_1''$ and $b_1'$ curves start to cross together and then start to stabilize.




Chapter 3. Figure 5 $\boldsymbol{\delta}$ value evolving pattern

According to Figure 5, from 61st iteration to 142nd iteration, several good potential good ADP policies are saved when implementing the DP loop. However, due to the setting that $m$ is equal to 5 and $\xi$ is equal to 0.005, the DP iteration stops at 137th iteration with almost 69 hours. And then, 137th aFVF is selected and we apply it to the simulation to investigate how it performs.

### 5.3. Simulation Results

In the simulation part, we apply the selected aFVF to the simulated environment in order to explore its performance. Moreover, greedy algorithm as a myopic algorithm used in Chen et al. (2017a, 2017b), is utilized as a benchmark in this study. At the initial stage, we set the battery inventory of each station as minimum level. As mentioned above, the demand profile in Fig. 4 is used in the simulation. There are only 96 periods in Fig. 4, thereby, we circulate as many as needed in the simulation. Following the simulation process proposed in Chen et al. (2017a), 15 scenarios are conducted with 200 stages for each scenario. Since we set the battery inventory of each station as minimum level at time period 1, for the DP simulation, the final results will be affected by the transient states (Taha, 2003). Therefore, in this study, we only consider the recurrent states in the simulation. After observing the dynamics of battery charging and

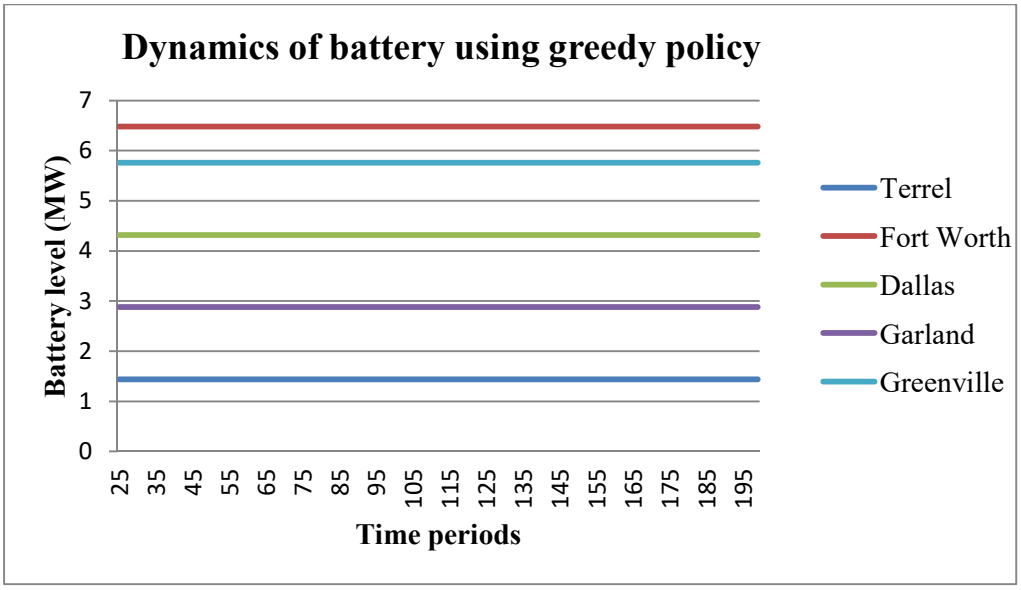
discharging within the 200 stages, the states from 25th stage to 200th stage are regarded as recurrent states. From Table 5, the simulation results of these 15 scenarios using the greedy policy and 137th aFVF. The values in Table 5 are calculated with the following Eq. (16). The values in Table 5 are estimated total costs from 25th stage to 200th stage using greedy policy and ADP policy. All estimated costs of 15 scenarios are much smaller than those of greedy policy, which indicates ADP policy is much better than greedy policy since the objective function is to minimize the operational cost. Fig. 6 shows an example of dynamics of battery in scenario 1 using greedy policy and ADP policy.

$$\min_{u_1,\ldots,u_t} E\{\sum_{t=25}^{200} \gamma^t c(s_t, u_t, \varepsilon_t)\} \tag{16}$$

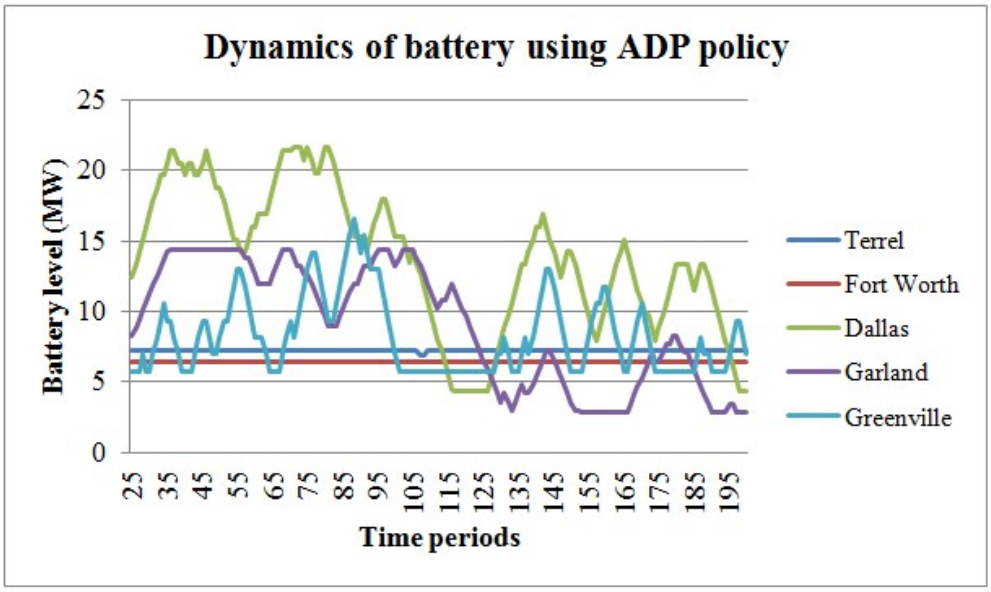
Table 3. 5 Estimated total costs of 15 scenarios using greedy policy and 137th aFVF in simulation

(Unit is thousands $)

| Order of Scenarios | Greedy policy | 137th aFVF |
|---|---|---|
| 1 | 913.14 | -1036.46 |
| 2 | 627.32 | -1097.76 |
| 3 | 990.12 | -1527.15 |
| 4 | 1087.69 | -835.44 |
| 5 | 1161.63 | -1008.51 |
| 6 | 936.19 | -1032.07 |
| 7 | 1424.36 | -391.89 |
| 8 | 995.70 | -1333.42 |
| 9 | 1564.57 | 57.35 |
| 10 | 905.70 | -580.08 |
| 11 | 1456.60 | -240.29 |
| 12 | 1200.74 | -679.61 |
| 13 | 1474.57 | 670.79 |
| 14 | 1338.57 | -536.73 |
| 15 | 1230.17 | -12.25 |




(a)

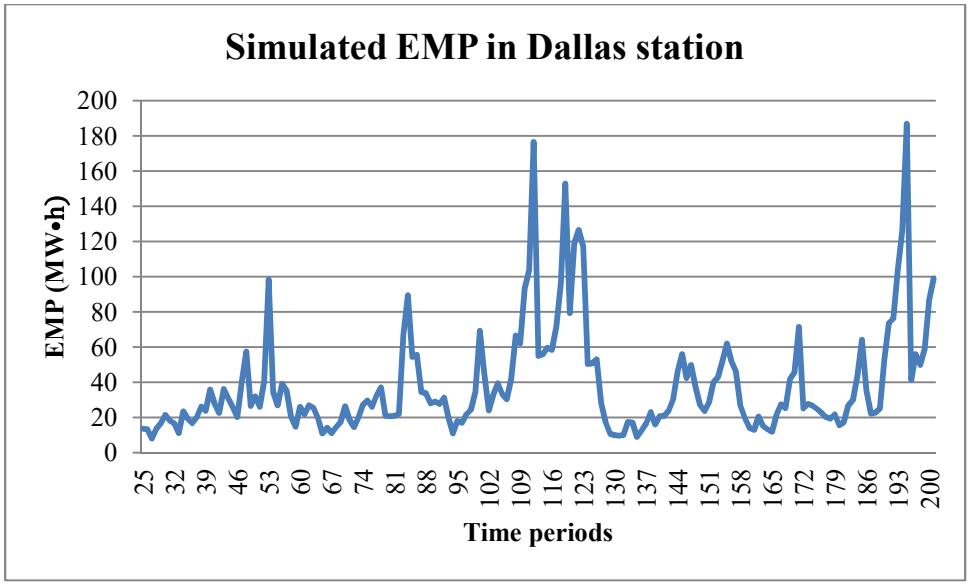



(b)

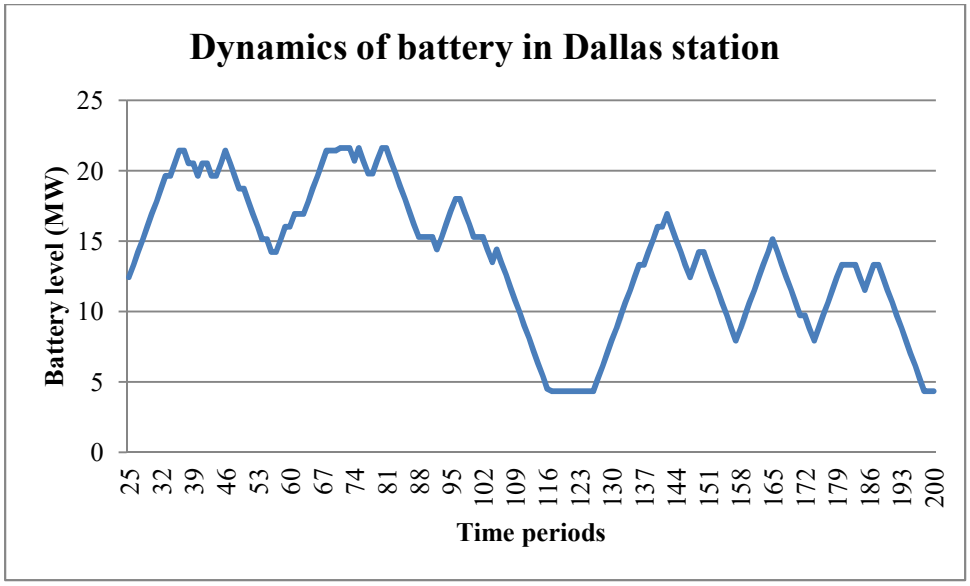
Chapter 3. Figure 6 Dynamics of battery using greedy policy in (a) and using ADP policy in (b). (Note: Terrel, Fort Worth, Dallas, Garland, Greenville refers to station 2, 3, 5, 7, 10 in Fig. 4, respectively.)

The difference between Fig. 6(a) and 6(b) is significant. With greedy policy, batteries in five stations have no charging and discharging behaviors. The reason for this is that if EMP is not negative, the batteries will not have any dynamic behaviors since greedy policy only seeks for local optimization for each stage without considering the future state. In another words, if the batteries are charged even with very low positive EMP, more cost will be added into the objective function, which is against the greedy policy itself. When checking simulated EMPs in all 15 scenarios, all EMPs are positive, which explains non-dynamics of batteries in Fig. 6(a)

Compared to Fig. 6(a), with ADP policy, except Fort Worth station, all those four stations have behaviors for charging and discharging. In order to present the dynamic control more clearly, the simulated EMP from Dallas station and the corresponding dynamics of battery using ADP policy is shown below:




(a)




(b)

Chapter 3. Figure 7 Simulated EMP in station 5 in (a) and its corresponding dynamics of battery in (b).

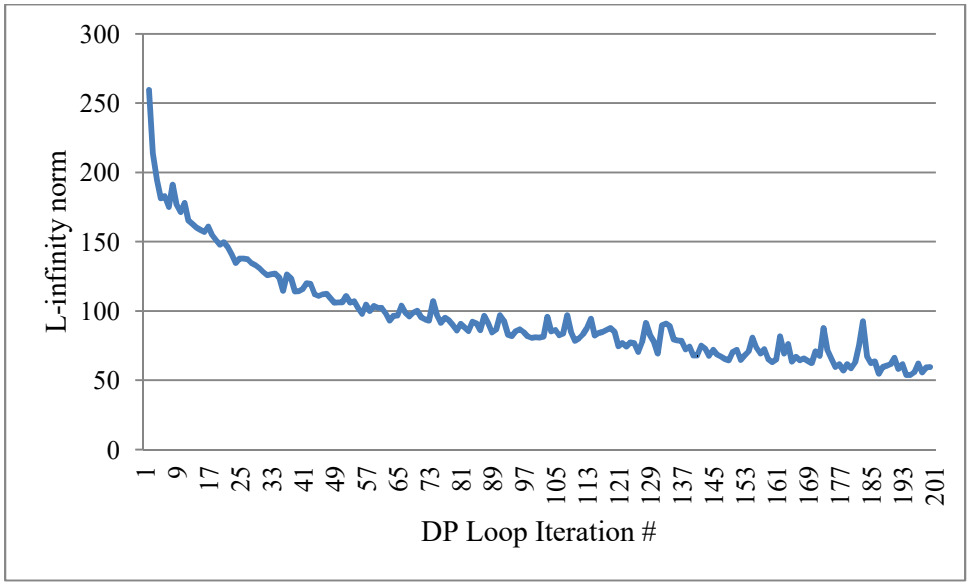
From Fig. 7, it is easy to find out that when the EMP is low, battery is charged and when the EMP is high, EMP is discharged, which indicates the ADP policy is able to control the system as expected. Based on the results from 137$^{th}$ aFVF and greedy policy, we can conclude ADP policy performs much better than greedy policy. It is noted that in the given example shown in Fig. 6 and Fig. 7, the first-type control strategy is not used since all EMPs are positive and all EMPs are no more than 200. However, the first type-control strategy may be used in the other scenarios.

## 6. Discussion

Before concluding the results, we further conduct another experiment in order to demonstrate how good the identified aFVF is but not just compare the 137$^{th}$ aFVF policy to greedy policy which is a relatively inferior one. In this further experiment, we employ the L-infinity norm stopping rule proposed by Powell (2007) to select an ADP policy for a comparison. This criterion is shown below:

$$\|V_k - V_{k-1}\| < \frac{\theta(1-\gamma)}{2\gamma}. \qquad (17)$$

The stopping criterion is reached when the maximum change in the value of any state is lower than the setting of right-hand side in Eq. (17), where $\gamma$ is the discount factor, and $\theta$ is a specified error tolerance. With the use of discount factor (0.995), the error tolerance value should be 3980 if we want the right-hand side in Eq. (17) equal to 10. However, the DP iteration is still continued after 200 iterations costing 4 days. Then, we plot the pattern of this L-infinity norm evolving pattern within 200 iterations as shown below.




Chapter 3. Figure 8 L-infinity norm value evolving pattern within 200 DP iterations

In Fig. 8, L-infinity norm evolving pattern is still going down slowly as observed, which obeys the value iterations based ADP algorithm and indicates the aFVF is on the way of convergence. According to Fig. 8, all the norm values are more than 50, which denotes that more DP iterations should be executed in order to reach this quantified rule. This experiment also indicates that on the way to the convergence of the aFVF, the error tolerance value should be set differently for different problems. However, before implementing ADP algorithm, it is hard to justify what error tolerance value should be set in advance. Compared to L-infinity norm stopping criteria, even though 45-degree line correspondence stopping rule also need to quantify the parameters ($\epsilon, \xi$ and

*m*), coordination of these three parameters is easier for it to capture the shape of the value function and stop the DP iteration reasonably in this case.

Table 3. 6 Estimated total costs of 15 scenarios using 137$^{th}$ aFVF and 200$^{th}$ aFVF in simulation

(Unit is thousands $)

| Order of Scenarios | 137$^{th}$ aFVF | 200$^{th}$ aFVF |
|---|---|---|
| 1 | -1036.46 | -1058.14 |
| 2 | -1097.76 | -1011.96 |
| 3 | -1527.15 | -1505.09 |
| 4 | -835.439 | -1064.68 |
| 5 | -1008.51 | -848.721 |
| 6 | -1032.07 | -1120 |
| 7 | -391.889 | -167.266 |
| 8 | -1333.42 | -1113.03 |
| 9 | 57.34632 | 157.2903 |
| 10 | -580.075 | -474.113 |
| 11 | -240.294 | -315.418 |
| 12 | -679.609 | -594.316 |
| 13 | 670.7892 | 647.2279 |
| 14 | -536.73 | -558.852 |
| 15 | -12.2482 | 80.00674 |

Based on the value iteration algorithm, on the way of convergence of value function, the later policy should be better or equal to the previous one. Therefore, we stop the DP iteration at 200$^{th}$ iteration and select the 200$^{th}$ aFVF as another benchmark, randomly and then execute it to the simulation in order to investigate its performance compared to that of 137$^{th}$ aFVF. In the above table, it is the simulation result. Estimated total costs for 176 periods for 15 scenarios are presented with different ADP policies. After conducted paired *t*-test used in Chen et al. (2017a and 2017b), since the *p*-value is 0.201, there are no statistical differences between 137$^{th}$ aFVF and 200$^{th}$ aFVF when α is given as 0.01 or even as high as 0.05. Based on this result, we can conclude that the quantified 45-degree line correspondence stopping criteria in Chen et al. (2017b) is able to stop the DP iteration reasonably and early, and select the ADP policy that has the same

performance statistically as the later one, which indicates a lot of computational time will be saved.

## 7. Conclusion

In this research, we apply the DACE based infinite horizon ADP algorithm proposed in Chen et al (2017a) to solve a large-scale, high-dimensional, infinite horizon, EV charging station control problem over a continuous state and decision space. In this system, renewable energy as resources provides electricity as well as main grid. In order to solve this control problem, first, we formulate it as a MDP problem, and then take advantage of MMFE and SVR forecasting models to formulate the transition model for the state variables such as wind power, solar power and EMP. For the wind speed, temperature and load profile, we use fixed trajectory from the historical data to execute the transition. Therefore, if assuming five stations open, there are 31 state variables in the state space, which indicates this is a much higher dimensional DP problem than previous research. For such large-scale DP problem, computational time is a significant challenge. Following the in research in Chen et al. (2017a and 2017b), we take advantage of 45-degree line correspondence stopping criterion to stop the DP iterations after quantifying its parameters. The selected $137^{th}$ ADP policy presents a significantly better performance than greedy policy as shown in Table 5. In addition, from Fig 6, and Fig. 7, ADP policy performs the way as we expected: when the EMP is low, the stations start to charge electricity and when the EMP is high, stored EMP in battery will be sold back to main grid. Furthermore, we try to use L-infinity norm criterion to select a ADP policy as another benchmark. From the L-infinity norm value evolving plot shown in Fig. 8, a downward pattern is observed, which indicates the aFVF is on the way of converging. However, if the error tolerance value is set to 10, more DP iterations need to be implemented. Therefore, we stop DP iteration at $200^{th}$ iteration and select the $200^{th}$ ADP policy as a comparison. Based on the concept of value iteration algorithm, on the way to the convergence of aFVF, the later ADP policy should perform better than or equal to the previous ones. After conducting a paired t-test on the simulation results, these two policies have the same performance statistically, which indicates the quantified 45-degree line correspondence stopping rule is able to stop the DP iteration reasonably and select a good solution. In the future, multicollinearity among the state variables should be resolved since this issue will result in transition models unstable. In this research, we make use of fixed trajectories from the historical

data to represent temperature, wind speed and load profile when building the aFVF. In the future, a related transitioned model should be developed.

**Acknowledgement**

This research project was supported in part by National Science Foundation Grant ECCS-1128871.

**Reference**

Ashtari, A., Bibeau, E., Shahidinejad, S., Molinski, T. PEV charging profile prediction and analysis based on vehicle usage data. IEEE Transactions on Smart Grid, 3(1): 341-350, 2012.

Badawy, M. O., Sozer, Y., Power flow management of a grid tied PV-battery system for electric vehicles charging. IEEE Transactions on Industry Applications. 53(2): 1347-1357, 2017.

Boaro, M., Fuselli, D., De Angelis, F., Liu, D., Wei, Q., Piazza, F. Adaptive Dynamic Programming Algorithm for Renewable Energy Scheduling and Battery Management. Cognitive Computation. 5(2): 264-277. 2013.

Chen, V. C. P., Ruppert, D., Shoemaker, C. A. Applying experimental design and regression splines to high-dimensional continuous-state stochastic dynamic programming. Operations Research, 47(1): 38-53, 1999.

Chen, Y., Li, H., Jin, K., Song, Q. Wind farm layout optimization using genetic algorithm with different hub height wind turbines, Energy Conversion and Management, Vol. (70) 56-65, 2013.

Chen, Y., Liu, F., Kulvanitchaiyanunt, A., Chen, V. C. P., Rosenberger, J., Wang, S. Infinite Horizon Approximate Dynamic Programming Using Computer Experiments. COSMOS 17-02, University of Texas at Arlington, 2017(a).

Chen, Y., Liu, F., Chen, V. C. P., Rosenberger, J. Application of Support Vector Regression into Stochastic Infinite Horizon Dynamic Program. COSMOS 17-03, University of Texas at Arlington, 2017(b).

Clement-Nyns, K., Haesen, E., Driesen, J. The impact of charging plug-in hybrid electric vehicles on a residential distribution grid. IEEE Transactions on Power Systems, 25(1): 371-380, 2010.

Cervellera, C., Chen, V. C. P., Wen, A. Optimization of a large-scale water reservoir network by stochastic dynamic programming with efficient state space discretization. European Journal of Operational Research, 171(3): 1139-1151, 2006.

Drucker, H., Burges, C. J. C., Kaufman, L., Smola, A. J., Vapnik, V. N. Support Vector Regression Machines, Advances in Neural Information Processing Systems 9, NIPS 1996, 155–161, MIT Press, 1997.

De Brabanter, K., Karsmakers, P., Ojeda, F., Alzate, C., De Brabanter, J., Pelckmans, K., De Moor, B., Vandewalle, J., Suykens, J. A. K. LS-SVM lab Toolbox User's Guide version 1.8. ESAT-SISTA Technical Report 10-146, August, 2011.

Ernst, D., Glavic, M., Capitanescu, F., Wehenkel, L. Reinforcement learning versus model predictive control: A comparison on a power system problem. IEEE Transactions on Systems, Man, and Cybernetics-Part B: Cybernetics; 39(2): 517-529, 2009.

Fell, K., Huber, K., Zink, B., Kalisch, R., Forfia, D., Hazelwood, D., Dang, N., Gionet, D., Musto, M., Johnson, W., Assessment of plug-in electric vehicle integration with ISO/RTO systems,*KEMA, Inc. and ISO/RTO Council,* 2010.

Fan, H., Tarun, P. K., Chen, V. C. P. Adaptive value function approximation for continuous-state stochastic dynamic programming. Computers & Operations Research, 40(4): 1076-1084, 2013.

Gan, L., Topcu, U., Low, S. H. Optimal decentralized protocol for electric vehicle charging. IEEE Transactions on Power Systems. 28(2): 940-951, 2013.

Guo, Y., Hu, J., Su, W. Stochastic optimization for economic operation of plug-in electric vehicle charging stations at a municipal parking deck integrated with on-site renewable energy generation, in Transportation Electrification Conference and Expo (ITEC), 2014 IEEE: 1-6, 2014.

Halton, J. H. On the efficiency of certain quasi-random sequences of points in evaluating multi-dimensional integrals. Numerische Mathematik, 2: pp. 84-90, 1960.

Heath, D., Jackson, P. Modeling the evolution of demand forecasts with application to safety stock analysis in production/distribution systems. IIE Transaction, 26(3): 17-30, 1994.

Jiang, D. R., Powell, W. B. Optimal hour-ahead bidding in the real-time electricity market with battery storage using approximate dynamic programming. INFORMS Journal on Computing. 27(3): 525-543, 2015.

Khodayar, M. E., Lei, W., Shahidehpour, M. Hourly coordination of electric vehicle operation and volatile wind power generation in SCUC, IEEE Transactions on Smart Grid: 3(3): 1271-1279, 2012.

Khosrojerdi, A., Xiao, M., Sariprueck, P., Allen, J., Mistree, F., Designing a system of plug-in hybrid electric vehicle charging stations, IDETC/CIE 2013, Portland, Oregon, USA, August 4-7, 2013.

Kulvanitchaiyanunt, A., Rosenberger, J., Lee, W., Chen, V., Sarikprueck, P. , A Linear Program for Control of a System of PHEV Charging Stations, IEEE Transactions on Industry Applications. 52(3): 2046-2052, 2016.

Kutner, M. H., Nachtsheim, C. J. Neter, J., Li, W. Applied Linear Statistical Models. Fifth Ed. McGraw-Hill Irwin, 2004.

Lee, J., Lee, J. H. Approximate dynamic programming strategies and their applicability for process control: a review and future directions. International Journal of Control, Automation, and Systems, 2(3): 263-278, 2004.

Marano, V., Rizzoni, G. Energy and economic evaluation of PHEVs and their interaction with renewable energy sources and the power grid. IEEE International Conference on Vehicular Electronics and Safety, 2008. ICVES 2008: 84-89, 2008.

Peng, X., Berseth, G., van de Panne, M. Terrain-adaptive locomotion skills using deep reinforcement learning. ACM Transactions on Graphics (TOG). 35(4): No. 81, 2016.

Powell, W. B. Approximate dynamic programming: solving the curses of dimensionality. John Wiley, New York, 2007.

Sarikprueck, P., Lee, W., Kulvanitchaiyanunt, A., Chen, V., Rosenberger, J. Bounds for optimal control of a regional plug-In electric vehicle charging station system. IEEE Industrial and Commercial Power Systems Technical Conference, Niagara Falls, ON, Canada, May 6-11, 2017.

Sarikprueck, P. Forecasting of wind, PV generation, and market price for the optimal operations of the regional PEV charging stations, Ph.D. dissertation. University of Texas at Arlington, 2015.

Shaaban, M. F., Ismail, M., El-Saadany, E. F., Zhuang, W. Real-time PEV charging/discharging coordination in smart distribution systems. IEEE Transactions on Smart Grid, vol. 5, pp. 1797-1807, 2014.

Shi, G., Wei, Q., Liu, D. Optimization of electricity consumption in office buildings based on adaptive dynamic programming. Soft Computing. pp 1-11, 2016.

Sobol, I. M. The distribution of points in a cube and the approximate evaluation of integrals. USSR Computational Mathematics and Mathematical Physics, 7: 784-802, 1967.

Steen, D., Tuan, L., Carlson O., Bertling, L., Assessment of electric vehicle charging scenarios based on demographical data. IEEE Transactions on Smart Grid, 3(3):1457-1468, 2012.

Sun, L., Wang, X., Liu, W., Lin, Z., Wen, F., Ang, S. P. *Salam, M. A.* Optimisation model for power system restoration with support from electric vehicles employing battery swapping. IET Generation, Transmission & Distribution. 10(2): 771-779, 2016.

Suykens, J. A. K., Van Gestel, T., De Brahanter, J., De Moor, B., Vandewalle, J. Least squares support vector machines. World Scientific Pub. Co. Singapore. 2002.

Taha, H. Operations research: an introduction (seventh ed.), Prentice Hall, New Jersey, 2003.

Tang, Y., He, H., Ni, Z., Wen, J., Sui, X. Reactive power control of grid-connected wind farm based on adaptive dynamic programming. Neurocomputing. 125(11): 125-133, 2014.

Wang, M., Ismail, M., Shen, X., Serpedin, E., Qaraqe, K. Spatial and temporal online charging/discharging coordination for mobile PEVs. IEEE Transactions on Wireless Communications. 22(1):112-121, 2015.

Wei, Q., Liu, D., Shi, G. A Novel Dual Iterative Q-Learning Method for Optimal Battery Management in Smart Residential Environments. IEEE Transactions on Industrial Electronics: 64(4), 2015.

Wei. Q., Liu, D., Lewis, F. L., Liu, Y., Zhang, J. Mixed Iterative Adaptive Dynamic Programming for Optimal Battery Energy Control in Smart Residential Microgrids. IEEE Transaction on Industrial Electronics. 64(5): 4110-4120, 2017.

Xie, S., Zhong, W., Xie, K., Yu, R., Zhang, Y. Fair energy scheduling for vehicle-to-grid networks using adaptive dynamic programming. IEEE Transaction on Neural Networks and Learning Systems. 27(8): 1697-1707, 2016.

Yao, L., Lim, W. H., Tsai, T. S. A real-time charging scheme for demand response in electric vehicle parking station. IEEE Transactions on Smart Grid, 8(1): 52-62, 2017.

Yu, R., Zhong, W., Xie, S., Zhang, Y., Zhang, Y. QoS Differential scheduling in cognitive-radio-based smart grid networks: an adaptive dynamic programming approach. IEEE Transaction on Neural Networks and Learning Systems. 27(2): 435-443, 2016.

Yang, Z., Chen, V. C. P., Chang, M. E., Sattler, M. L., Wen, A. A decision making framework for ozone pollution control. Operations Research, 57(2): 484 498, 2009.

Zhu, Z., Lambotharan, S., Chin, W. H., Fan, Z. A mean field game theoretic approach to electric vehicles charging. *IEEE Access,* 4: 3501-3510, 2016.

## Chapter 4. Conclusion

In this dissertation, we focus on solving high-dimensional, infinite horizon DP problems over a continuous space. Due to the "curse of dimensionality", especially in a continuous space, we further develop a DACE based infinite horizon ADP algorithm proposed in Chen et al. (2017) to conduct the research. In this dissertation, there are two components of the research: one is in methodology and the other one is an application. In the methodology part (Chapter 3), we employ support vector regression (SVR) to approximate the value function considering the success that SVR has achieved in data mining. Compared to the performance made by MARS in Chen et al. (2017), SVR demonstrates steadier behavior in value function approximation. In addition, we detail the 45 degree line correspondence stopping criterion to stop the DP algorithm when the shape of value function is identified, which was proposed in Chen et al. (2017) . Moreover, after simulating the ADP policies, the selected ADP policies have a much better performance than a benchmark greedy policy, and the selected ADP policy by 45 degree line correspondence has the same performance as the one stopped by the L-infinity norm, which indicates the specification of the 45 degree line correspondence is reasonable. Furthermore, we explore the extrapolation of the value function created by SVR. From the 3D meshplot, when enlarging the plot range, the extrapolation error is observed when using SVR. However, after implementing a closer experiment, we find out the extrapolation error in that problem is not significant.

After this research, we apply DACE based infinite horizon ADP algorithm with SVR to solving a large-scale, high-dimensional, infinite horizon, EV charging station control problem over a continuous space. In this system, renewable energy as sources provides electricity as well as main grid. Therefore, if assuming five stations are open, there are 31 state variables in the state space, which indicates this is a much higher dimensional problem than the work in the literature. Considering this, first of all, we formulate a Level 3 fast DC charging station system as a Markov decision process problem. And then based on forecasting models proposed in Sarikprueck (2015), we develop the transition model for each state vector. For such large-scale DP problem, computational time is a significant challenge. With the 45 degree line correspondence stopping rule, the $137^{th}$ aFVF is identified as a high-quality policy. Then, we apply it to the simulated environment. The simulation results clearly demonstrate the benefits of the ADP policy vs. greedy policy: the greedy policy has no change in battery level when the EMPs are positive, but the ADP policy is able to buy more electricity when the EMP is low, store the surplus, and sell

the stored electricity back to power grid when the EMP is high. Moreover, a paired t-test is conduct between the simulation results using the $137^{th}$ aFVF and the $200^{th}$ aFVF. Since the *p*-value is 0.201, we conclude these two ADP policies have the same performance statistically, which shows that 45 degree line correspondence rule is able to identify a high-quality solution at an early DP iteration. In the future, multicollinearity among the state variables should be resolved, since this could result in unstable transition models. Another research should focus on developing a general rule to select the error tolerances in the stopping criteria (L-infinity norm and 45 degree line correspondence).

**Reference**

Chen, Y., Liu, F., Kulvanitchaiyanunt, A., Chen, V. C. P., Rosenberger, J., Wang, S. Infinite Horizon Approximate Dynamic Programming Using Computer Experiments. COSMOS's 17-02, University of Texas at Arlington, 2017.

Sarikprueck, P. Forecasting Of Wind, PV Generation, And Market Price For The Optimal Operations Of The Regional PEV Charging Stations, Ph.D. dissertation. University of Texas at Arlington, 2015.