University of Strathclyde

Department of Electronic and Electrical Engineering

Learning to Trade Power

by

Richard W. Lincoln

A thesis presented in fulfilment of the
requirements for the degree of

Doctor of Philosophy

2010

This thesis is the result of the author's original research. It has been composed by the author and has not been previously submitted for examination which has led to the award of a degree.

The copyright of this thesis belongs to the author under the terms of the United Kingdom Copyright Acts as qualified by University of Strathclyde Regulation 3.50. Due acknowledgement must always be made of the use of any material contained in, or derived from, this thesis.

Signed: Date: September 5, 2010

Acknowledgements

Abstract

In Electrical Power Engineering, learning algorithms can be used to model the strategies of electricity market participants. The objective of this work is to establish if *policy gradient* reinforcement learning algorithms can be used to create participant models superior to those involving previously applied *value function* based methods.

Supply of electricity involves technology, money, people, natural resources and the environment. All of these aspects are changing and electricity market designs must be suitably researched to ensure that they are fit for purpose. In this thesis electricity markets are modelled as non-linear constrained optimisation problems, which are solved using a primal-dual interior point method. Policy gradient reinforcement learning algorithms are used to adjust the parameters of multi-layer feed-forward artificial neural networks that approximate each market participant's policy for selecting power quantities and prices that are offered in the simulated marketplace.

Traditional reinforcement learning methods, that learn a value function, have been previously applied in simulated electricity trade, but they are mostly restricted to use with discrete representations of a market environment. Policy gradient methods have been proven to offer convergence guarantees in continuous environments and avoid many of the problems that mar value function based methods.

Five types of learning algorithm are compared in a series of Nash equilibrium and constraint exploitation simulations. Policy gradient methods are found to be a valid option for modelling the strategies of electricity market participants, but they are outperformed by a traditional action-value function algorithm in all of the tests. Further development of this research could provide opportunities for advanced learning algorithms to be used in descision support and automated energy trade applications.

Contents

List of Figures

List of Tables

Chapter 1

Introduction

This thesis examines reinforcement learning algorithms in the domain of electric power trade. In this chapter the motivation for research into electricity trade is explained, the problem under consideration is defined and the principle research contributions are stated.

1.1 Research Motivation

Quality of life for a person is directly proportional to his or her electricity usage (Alam, Bala, Huo, & Matin, 1991). The world population is currently 6.7 billion and forecast to exceed 9 billion by 2050 (United Nations, 2003). Electricity production currently demands over one third of the annual primary energy extracted (The International Energy Agency, 2010) and as people endeavour to improve their quality of life, finite fuel resources will become increasingly scarce. Market mechanisms, such as auctions, where the final allocation is based upon the claimants' willingness to pay for the goods, provide a device for efficient allocation of resources in short supply.

Commercialisation of large electricity supply industries began two decades ago in the UK. The inability to store electricity, once generated, in a commercially viable quantity prevents trade as a conventional commodity. Trading mechanisms must allow shortfalls in electric energy to be purchased at short notice from quickly dispatchable generators. Designed correctly, a competitive electricity market promotes efficiency and drives down costs to the consumer, while design errors can allow market power abuse and elevated market prices.

The average total demand for electricity in the United Kingdom (UK) is approximately 45GW and the cost of buying 1MW for one hour is around £40 (Department of Energy and Climate Change, 2009). This equates to yearly trans-

action values of £16 billion. The value of electricity to society is particularly apparent when supply fails. The New York black-out in August 2003 involved a loss of 61.8GW of power supply to approximately 50 million consumers. The majority of supplies were restored within two days, but the event is estimated to have cost more than $6 billion (Minkel, 2008; ICF Consulting, 2003).

The value of electricity to society makes it impractical to experiment with radical changes to trading arrangements on real systems. An alternative is to study abstract mathematical models with sets of simplifying approximations and assumptions and, where possible, to find analytical solutions using digital computer programs. Competition is a fundamental part of all markets, but the strategies of human participants are difficult to model. Reinforcement learning methods can be used to represent adaptive behaviour in competing players and are capable of learning complex strategies (Tesauro, 1994).

1.2 Problem Statement

Individuals participating in an electricity market (be they representing generating companies, load serving entities, firms of traders etc.) must utilise noisy, mostly continuous, multi-dimensional data to their advantage. Certain types of data, e.g. demand forecasts, are uncertain and other types, e.g. the bids of competitors, are hidden. Reinforcement learning algorithms must operate with data of this kind if they are to successfully model participant strategies.

Traditional reinforcement learning methods attempt to find the *value* of each available action in a given state. When discrete state and action spaces are defined, these methods become restricted by Bellman's Curse of Dimensionality (Bellman, 1961) and can not be applied to highly complex problems. When used with function approximation techniques (e.g. artificial neural networks) they can be applied to continuous representations of an environment. However, the greedy updates used by most techniques have been shown to cause algorithms approximating a value function to not converge or even diverge (Tsitsiklis & Roy, 1994; Peters & Schaal, 2008; Gordon, 1995; Baird, 1995).

Policy gradient reinforcement learning methods do not attempt to approximate a value function, but to approximate a *policy-function* that, given the current perceived state of the environment, returns an action. They do not suffer from many of the problems that mar value-function based methods in high-dimensional problems. They have strong convergence properties, do not require that all states be continuously visited and work with state and action spaces that

are continuous, discrete or mixed. Policy performance may be degraded by uncertainty in state data, but the learning methods do not need to be altered. They have been successfully applied in many operational settings, including: robotic control (Peters & Schaal, 2006), financial trading (Moody & Saffell, 2001) and network routing (Peshkin & Savova, 2002) applications.

It is proposed in this thesis that agents which learn using policy gradient methods may outperform those using value function based methods in simulated competitive electricity trade. It is further proposed that policy gradient methods may operate better under dynamic electric power system conditions, achieving greater profit by exploiting constraints to their financial benefit. This thesis will use electricity market simulation techniques to compare value function based and policy gradient learning methods to explore these proposals.

1.3   Research Contributions

The research presented in this thesis pertains to the academic fields of Electric Power Engineering, Artificial Intelligence and Economics. The principle contributions made by this thesis are:

- The first application of policy gradient reinforcement learning methods in simulated electricity trade.

- The first application of a non-linear optimal power flow formulation in agent based electricity market simulation.

- A new Stateful Roth-Erev reinforcement learning method.

- Simulation results comparing the convergence to a Nash equilibrium of policy gradient and value function based reinforcement learning methods.

- Simulation results that examine the exploitation of electric power system constraints by policy gradient reinforcement learning methods.

- An implementation of a power exchange auctions market model and multi-learning-agent system for simulating electricity trade.

- The idea of applying Neuro-Fitted Q-Iteration and $GQ(\lambda)$ in simulations of competitive energy trade.

- A model of the UK transmission system derived from data in the National Grid Seven Year Statement.

The publications that have resulted from this thesis are: Lincoln, Galloway, and Burt (2009, 2007); Lincoln, Galloway, Burt, and McDonald (2006).

1.4 Thesis Outline

The presentation of this thesis is organised into nine chapters. Chapter 2 provides background information on electricity supply, wholesale electricity markets and reinforcement learning. It describes how optimal power flow formulations can be used to model electricity markets and defines the reinforcement learning algorithms that are later compared.

In Chapter 3 the research in this thesis is described in the context of previous work that is related in terms of application field and methodology. Publications on agent based electricity market simulation are reviewed with emphasis on the reinforcement learning methods used. Previous applications of policy gradient learning methods in other types of market setting are reviewed also.

Chapter 4 describes the power exchange auction market model and the multi-agent system used to simulate electricity trade. It defines the association of learning agents with portfolios of generators, the process of offer submission and the reward process.

Simulations that examine the convergence to a Nash equilibrium of systems of multiple electric power trading agents is reported in Chapter 5. A six bus test case is used and results for four learning algorithms under two cost configurations are presented and analysed.

Chapter 6 examines the ability of agents to learn policies for exploiting constraints in simulated power systems. The 24 bus model from the IEEE Reliability Test System provides a complex environment with dynamic loading conditions.

The primary conclusions drawn from the results in this thesis are summarised in Chapter 7. Shortcomings of the approach are noted and the broader implications are addressed. Some ideas for further work are also outlined, including alternative reinforcement learning methods and uses for a model of the UK transmission system.

Chapter 2

Background

This chapter provides background information on electricity market and electric power system simulation. A brief introduction to national electricity supply is given along with a history of UK wholesale electricity market designs. Approaches to market simulation that include transmission system constraints are introduced and definitions for the learning algorithms, that are later used to model market participant behaviour, are provided.

2.1 Electric Power Supply

Generation and bulk movement of electricity in the UK takes place in a three-phase alternating current (AC) power system. The *phases* are high voltage, sinusoidal electrical waveforms, offset in time from each other by 120 degrees and oscillating at approximately 50Hz. Synchronous generators (sometimes known as alternators), typically rotating at 3000 or 1500 revolutions per minute, generate apparent power S at a line voltage V_l typically between 11kV and 25kV. One of the principal reasons that AC, and not direct current (DC), systems are common in electricity supply is that they allow power to be transformed between voltages with very high efficiency. The output from a power station is typically stepped-up to 275kV or 400kV for transmission over long distances. The apparent power conducted by a three-phase transmission line l is the product of the line current I_l and the line voltage:

$$S = \sqrt{3}V_l I_l \tag{2.1}$$

Therefore the line current is inversely proportional to the voltage at which the power is transmitted. Ohmic heating losses are directly proportional to the *square*

Transmission Lines 400kV, 275kV or 138kV

Generating Station   Generator Step-   Transmission Customer   Substation Step-
11kV–25kV   Up Transformer   275kV and 132kV   Down Transformer

Primary Customer
33kV, 11kV or 6.6kV

Generation: ....
Transmission: ....   Secondary Customer
Distribution: ——————   400V and 220V
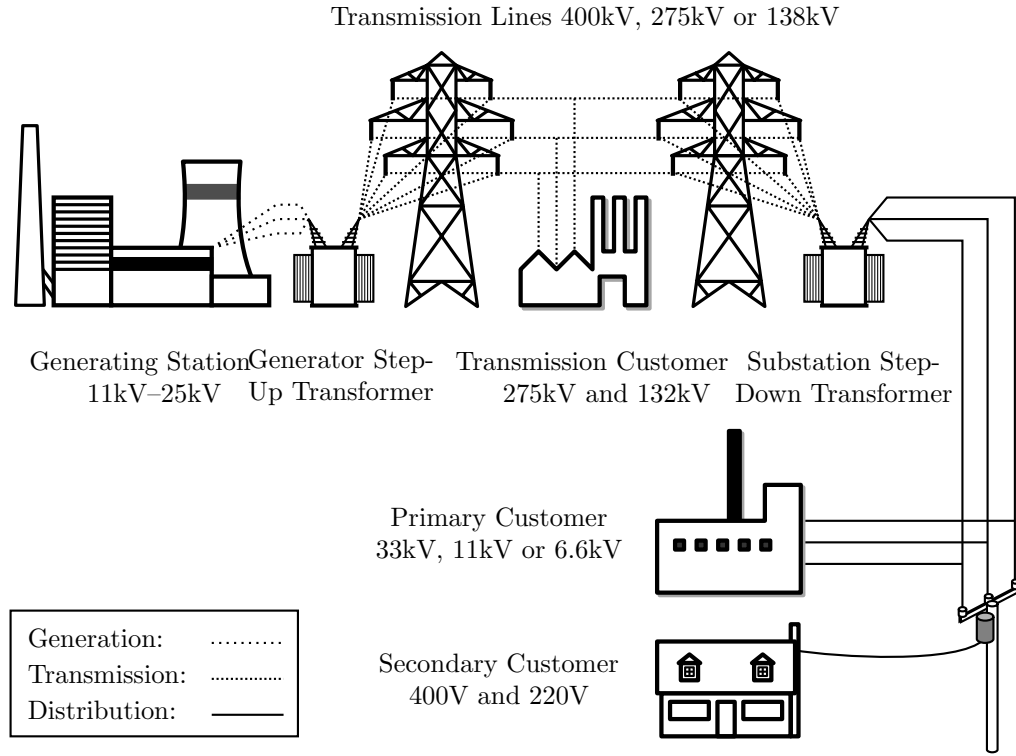
Figure 2.1: Basic structure of a three phase AC power system.

of the line current

$$P_r = 3I_l^2 R \tag{2.2}$$

where R is the resistance of the transmission line. Hence, any reduction in line current dramatically reduces the amount of energy wasted through heating losses. One consequence of high voltages is the larger extent and integrity of the insulation required between conductors, neutral and earth. This is the reason that transmission towers are typically large and undergrounding systems is expensive.

The UK transmission system operates at 400kV and 275kV (and 132kV in Scotland), but systems with voltages up to and beyond 1000kV are used in larger countries such as Canada and China (WG 31.04, 1983). For transmission over very long distances or undersea, high voltage DC (HVDC) systems have become economically viable in recent years. The reactance of a transmission line is proportional to frequency so one advantage of an HVDC system is that the reactive power component in is nil and more active power flow can be transmitted in a line/cable of a certain diameter.

The ability to transform power between voltages and transmit large volumes over long distances allows electricity generation to take place at high capacity power stations, which offer economies of scale and lower operating costs. It allows

electricity to be transmitted across country borders and from renewable energy plant, such as hydro-electric power stations, located in remote areas. Figure 2.2 shows how larger power stations in the UK are located away from load centres and close to sources of fuel, such as the coal fields in northern England and gas supply terminals near Cardiff and London.

For delivery to most consumers, electric energy is transferred at a substation from the transmission system to the grid supply point of a distribution system. Distribution networks in the UK are also three-phase AC power systems, but typically operate at lower voltages and differ in their general structure (or topology) from transmission networks. Transmission networks are typically highly interconnected, providing multiple paths for power flow. Distribution networks in rural areas typically consist of long radial feeders (usually overhead lines) and in urban areas, of many ring circuits (usually cables). Three-phase transformers, that step the voltage down to levels more convenient for general use (typically from 11kV or 33kV to 400V), are spaced out on the feeders/rings. All three-phases at 400V may be provided for industrial and commercial loads or individual phases at 230V supply typical domestic and other commercial loads. Splitting of phases is usually planned so that each is loaded equally. If achieved, this produces a balanced, symmetrical system with zero current flow on the neutral and it can be analysed as a *single* phase circuit (See Section Section 2.3.2 below). Figure 2.1 illustrates the basic structure of a typical national electric power system (U.S.-Canada Power System Outage Task Force, 2004).

2.2   Electricity Markets

The UK was the first large country to privatise its electricity supply industry when it did so in the early 1990s. The approach has been used as a model by other countries and the market structures that have since been implemented in the UK have used many of the main concepts for national electricity market design.

The England and Wales Electricity Pool was created in 1990 to break up the vertically integrated Central Electricity Generating Board (CEGB) and to gradually introduce competition in generation and retail supply. The Pool has since been replaced by trading arrangements in which market outcomes are not centrally determined, but arise largely from bilateral agreements between producers and suppliers.
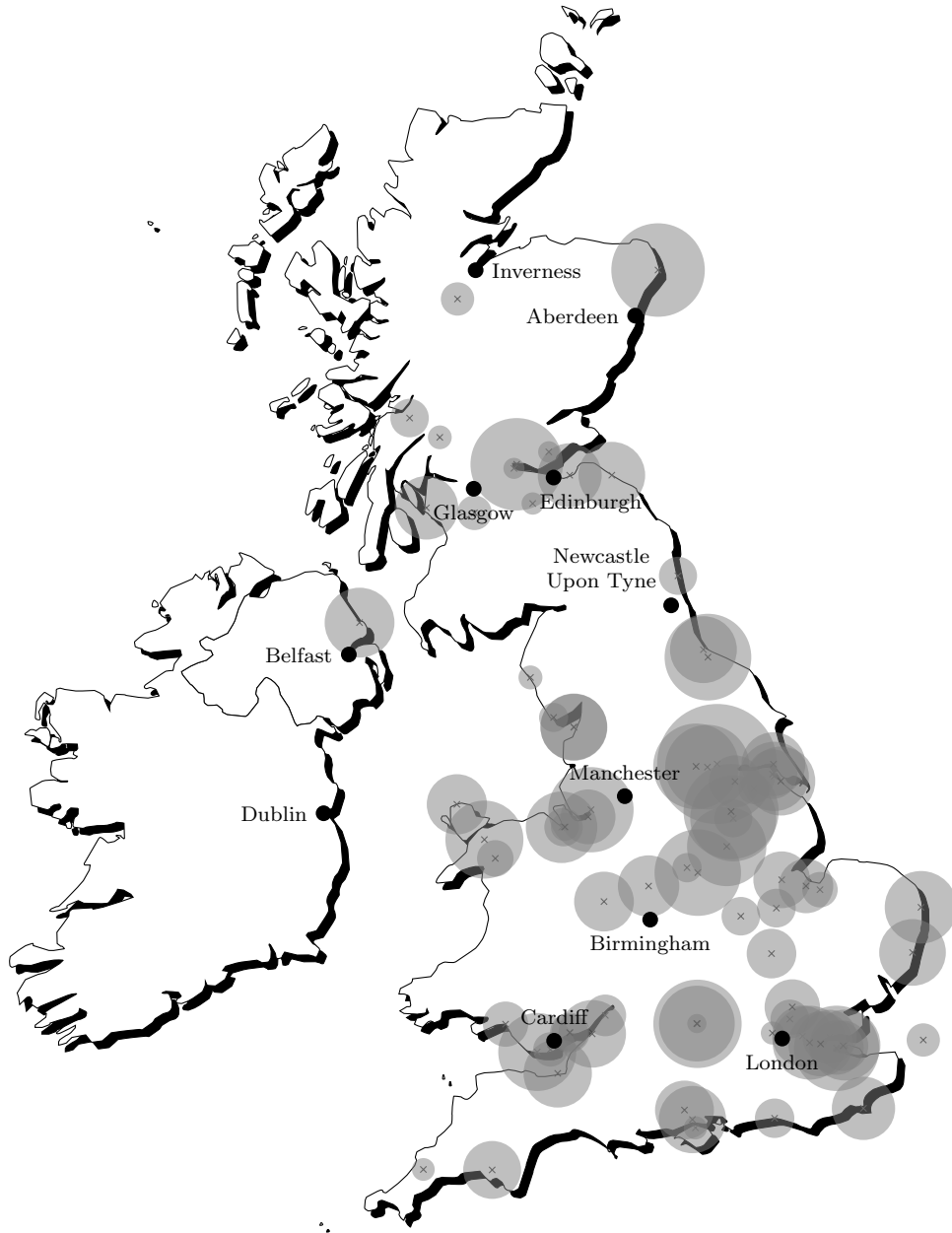
Figure 2.2: UK power station locations.

2.2.1 The England and Wales Electricity Pool

The Electric Lighting Act 1882 initiated the development of the UK's electricity supply industry by permitting persons, companies and local authorities to set up supply systems, principally at the time for the purposes of street lighting and trams. The Central Electricity Board started operating the first grid of interconnected regional networks (synchronised at 132kV, 50Hz) in 1933. This began operation as a national system five years later and was nationalised in 1947. Over 600 electricity companies were merged in the process and the British Electricity Authority was created. It was later dissolved and replaced with the CEGB and the Electricity Council under The Electricity Act 1957. The CEGB was responsible for planning the network and generating sufficient electricity until the beginning of privatisation.

The UK electricity supply industry was privatised, and The England and Wales Electricity Pool created, in March 1990. Control of the transmission system was transferred from the CEGB to the National Grid Company, which was originally owned by twelve regional electricity companies and has since become publicly listed. The Pool was a multilateral contractual arrangement between generators and suppliers and did not itself buy or sell electricity. Competition in generation was introduced gradually, by first entitling customers with consumption greater than or equal to 1MW (approximately 45% of the non-domestic market (Department of Energy and Climate Change, 2009)) to purchase electricity form any listed supplier. This limit was lowered in April 1994 to include customers with peak loads of 100kW or more. Finally, between September 1998 and March 1999 the market was opened to all customers.

Scheduling of generation was on a merit order basis (cheapest first) at a day ahead stage and set a wholesale electricity price for each half-hour period of the schedule day. Forecasts of total demand in MW, based on historic data and adjusted for factors such as the weather, for each settlement period were used by generating companies and organisations with interconnects to the England and Wales grid to formulate bids that had to be submitted to the grid operator by 10AM on the day before the schedule day.

Figure 2.3 illustrates four of the five price parameters that would make up a bid. A start-up price would also be stated, representing the cost of turning on the generator from cold. The no-load price c_0 represents the cost in pounds of keeping the generator running regardless of output. Three incremental prices c_1, c_2 and c_3 specify the cost in £/MWh of generation between set-points p_1, p_2 and p_3.
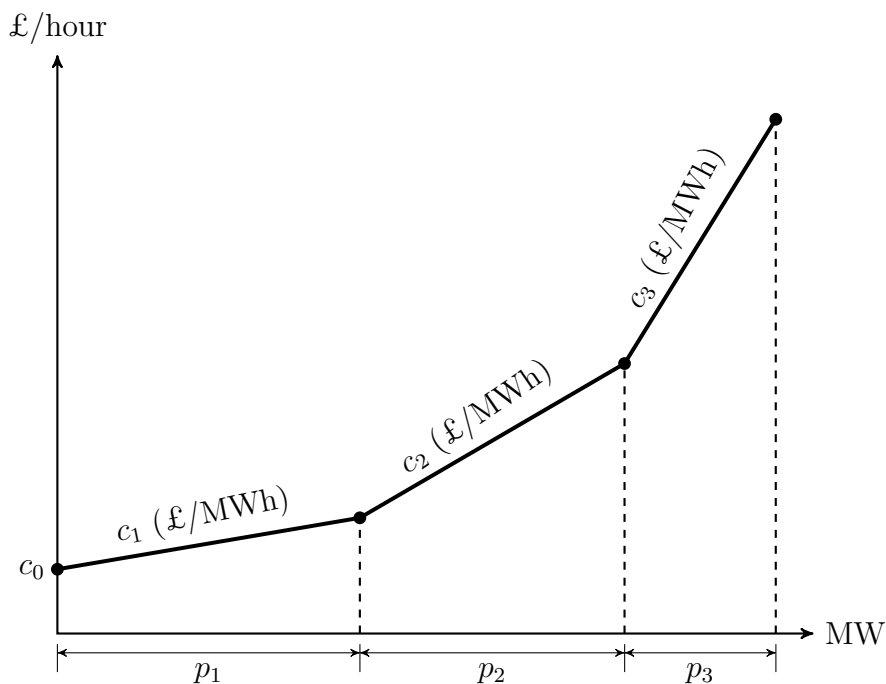
Figure 2.3: Pool bid structure.

A settlement algorithm would determine an unconstrained schedule (with no account being taken for the physical limitations of the transmission system), meeting the forecast demand and requirements for reserve while minimising cost. Cheapest bids up to the marginal point would be accepted first and the bid price from the marginal generator would generally determine the system marginal price for each settlement period. The system marginal price would form the basis of the prices paid by consumers and paid to generators, which would be adjusted such that the costs of transmission are covered by the market and that the availability of capacity is encouraged at certain times.

Variations in demand and changes in plant availability would be adjusted for by the grid operator between day close and physical delivery, producing a constrained schedule. Generators having submitted bids would be instructed to increase or reduce production as appropriate. Alternatively, the grid operator could instruct large customers with contracts to curtail their demand to do so or instruct generators contracted to provide ancillary services to adjust production.

2.2.2 British Electricity Transmission and Trading Arrangements

Concerns over the exploitation of market power in The England and Wales Electricity Pool and over the ability of the market to reduce consumer electricity prices prompted the introduction of New Electricity Trading Arrangements (NETA) in March 2001 (D. Bunn & Martoccia, 2005). The aim was to improve efficiency and provide greater choice to participants. Control of the Scottish transmission system was included with the introduction of the nationwide British Electricity Transmission and Trading Arrangements (BETTA) in April 2005 under The Energy Act 2004. While The Pool operated a single daily auction and dispatched plant centrally, under the new arrangements participants became self-dispatching and market positions became determined through continuous bilateral trading between generators, suppliers, traders and consumers.

The majority of power is traded under the BETTA through long-term contracts that are customised to the requirements of each party (Kirschen & Strbac, 2004). These instruments suit participants responsible for large power stations or those purchasing large volumes of power for many customers. Relatively, large amounts of time and effort are typically required for these long-term contracts to be initially formed and this results in a high associated transaction cost. However, they reduce risk for large players and often include a degree of flexibility.

Electric power is also traded directly between participants through over-the-counter contracts that usually have a standardised form. Such contracts typically concern smaller volumes of power and have lower associated transaction costs. Often they are used by participants to refine their market position ahead of delivery time (Kirschen & Strbac, 2004).

Trading facilities, such as power exchanges, provide a means for participants to fine-tune their positions further, through short-term transactions for often relatively small quantities of energy. Modern exchanges are computerised and accept anonymous offers and bids submitted electronically.

All bilateral trading must be completed before "gate-closure": a point in time before delivery that gives the system operator an opportunity to balance supply and demand and mitigate potential breaches of system limits. In keeping with the UK's free market philosophy, a competitive spot market (Schweppe, Caramanis, Tabors, & Bohn, 1988) forms part of the balancing mechanism. A generator that is not fully loaded may offer a price at which it is willing to increase its output by a specified quantity, stating the rate at which it is capable of doing so. Certain loads may also offer demand reductions at a price which can typically be

implemented very quickly. Longer-term contracts for balancing services are also struck between the system operator and generators/suppliers in order to avoid the price volatility often associated with spot markets.

2.3 Electricity Market Simulation

Previous sections have identified the importance of electricity to modern societies and explained how the majority of electricity supply in the UK is trusted to unadministered bilateral trade. All aspects of electricity supply are constantly changing and electricity markets must be suitably researched to ensure that their designs are fit for purpose. The value of electricity to society means that it is not practical to experiment with radical changes to trading arrangements on real systems.

Game theory is the branch of applied mathematics in which behaviour in strategic situations is captured mathematically. A common approach to doing this is to model the system and players as a mathematical optimisation problem. Optimal power flow is a classic optimisation problem in the field of Electrical Power Engineering and variants of it are widely used in electricity market research. In this thesis, optimal power flow forms part of an *agent-based* simulation: an alternative approach to the mathematics of games.

2.3.1 Agent-Based Simulation

Social systems, such as electricity markets, are inherently complex and involve interactions between different types of individual and between individuals and collective entities, such as organisations or groups, the behaviour of which is itself the product of individual interactions. This complexity drives classical monolithic equilibrium models to their limits. The models are often highly stylised and limited to small numbers of players with strong constraining assumptions made on their behaviour.

Agent-based simulation involves modelling the simultaneous operations of, and interactions between adaptive agents and then assessing their effect on the system as a whole. System properties arise from agent interactions, even those with simple behavioural rules, that could not be deduced by simply aggregating the agent's properties.

Following Tesfatsion and Judd (2006), the objectives of agent-based modelling research fall roughly into four strands: empirical, normative, heuristic and

methodological. The *empirical* objectives are to understand how and why macro-level regularities have evolved from micro-level interactions when little or no top-down control is present. Research with *normative* goals aims to relate agent-based models to an ideal standard or optimal design. The objective being to evaluate proposed designs for social policy, institutions or processes in their ability to produce socially desirable system performance. The *heuristic* strand aims to generate theories on the fundamental causal mechanisms in social systems that can be observed when there are alternative initial conditions. This thesis aims to provide *methodological* advancement. Improvements in the tools and methods available aids research with the former objectives.

2.3.2   Optimal Power Flow

Nationalised electricity supply industries were for many years planned, operated and controlled centrally. A system operator would determine which generators must operate and the required output of the operating units such that demand and reserve requirements were met and the overall cost of production was minimised. In Electric Power Engineering, these are termed the *unit commitment* and *economic dispatch* problems.

A formulation of the unit commitment problem was published in 1962 that incorporated electric power system constraints (Carpentier, 1962). Optimal power flow is this combination of the economic and power flow aspects of power systems into one mathematical optimisation problem. The ability of optimal power flow to solve centralised power system operation problems and determine prices in power pool markets has resulted in it becoming one of the most widely studied subjects in the electric power systems community.

Power Flow Formulation

Optimal power flow derives its name from the power flow (or load flow) steady-state power system analysis technique. Given sets of generator data, load data and a nodal admittance matrix, a power flow study determines the complex voltage

$$V_i = |V_i|\angle\delta_i = |V_i|(\cos\delta_i + j\sin\delta_i) \tag{2.3}$$

at each node i in the power system, from which line flows may be calculated (Grainger & Stevenson, 1994).

Nodal Admittance Matrix   The nodal admittance matrix describes the electrical network and its formulation is dependant upon the transmission line, transformer and shunt models employed. A branch in a nodal representation of a power system is typically modelled as a medium length transmission line in series with a regulating transformer at the "from" end (Zimmerman, 2010, p.11). A nominal-π model with total series admittance $y_s = 1/(r_s + jx_s)$ and total shunt capacitance b_c is often to represent the transmission line. The transformer may assumed to be ideal, phase-shifting and tap-changing, with the ratio between primary winding voltage v_f and secondary winding voltage $N = \tau e^{j\theta_{ph}}$ where τ is the tap ratio and θ_{ph} is the phase shift angle. Figure 2.4 diagrams this conventional branch model. From Kirchhoff's Current Law the current in the series impedance is

$$i_s = \frac{b_c}{2}v_t - i_t \tag{2.4}$$

and from Kirchhoff's Voltage Law the voltage across the secondary winding of the transformer is

$$\frac{v_f}{N} = v_t + \frac{i_s}{y_s} \tag{2.5}$$

Substituting i_s from equation (2.4), gives

$$\frac{v_f}{N} = v_t - \frac{i_t}{y_s} + v_t \frac{b_c}{2y_s} \tag{2.6}$$

and rearranging in terms if i_t, gives

$$i_t = v_s \left(\frac{-y_s}{\tau e^{\theta_{ph}}} \right) + v_r \left(y_s + \frac{b_c}{2} \right) \tag{2.7}$$

The current through the secondary winding of the transformer is

$$N^* i_f = i_s + \frac{b_c}{2} \frac{v_f}{N} \tag{2.8}$$

Substituting i_s from equation (2.4) again, gives

$$N^* i_f = \frac{b_c}{2} v_t - i_t + \frac{b_c}{2} \frac{v_f}{N} \tag{2.9}$$

and substituting $\frac{v_f}{N}$ from equation (2.6) and rearranging in terms if i_s, gives

$$i_s = v_s \left(\frac{1}{\tau^2} \left(y_s + \frac{b_c}{2} \right) \right) + v_r \left(\frac{y_s}{\tau e^{-j\theta}} \right) \tag{2.10}$$
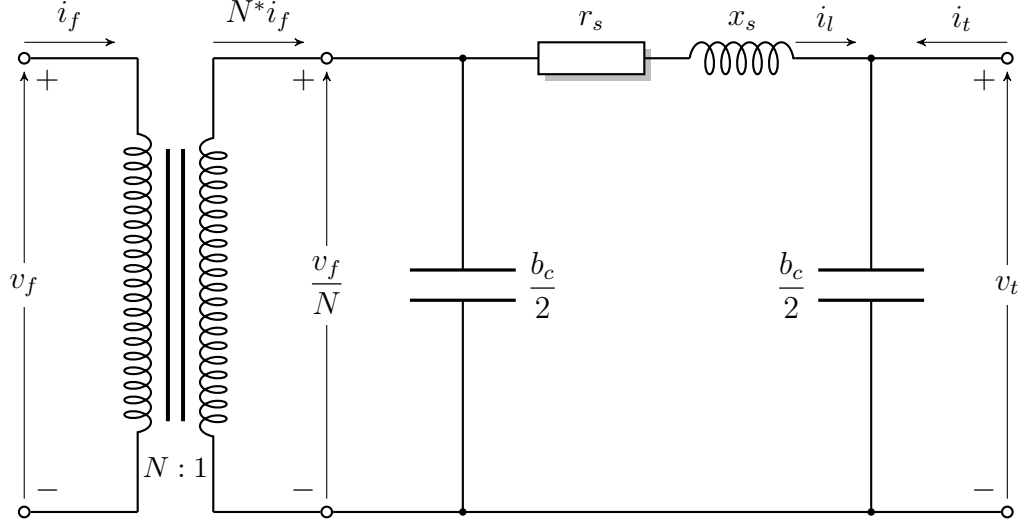
Figure 2.4: Nominal-π medium length transmission line model in series with a phase shifting, tap changing transformer model.

Combining equations (2.7) and (2.10), the *from* and *to* end complex current injections for branch l are

$$
\begin{bmatrix} i_f^l \\ i_t^l \end{bmatrix} = \begin{bmatrix} y_{ff}^l & y_{ft}^l \\ y_{tf}^l & y_{tt}^l \end{bmatrix} \begin{bmatrix} v_f^l \\ v_t^l \end{bmatrix}
\tag{2.11}
$$

where

$$
y_{ff}^l = \frac{1}{\tau^2} \left(y_s + \frac{b_c}{2} \right)
\tag{2.12}
$$

$$
y_{ft}^l = \frac{y_s}{\tau e^{-j\theta_{ph}}}
\tag{2.13}
$$

$$
y_{tf}^l = \frac{-y_s}{\tau e^{j\theta_{ph}}}
\tag{2.14}
$$

$$
y_{tt}^l = y_s + \frac{b_c}{2}
\tag{2.15}
$$

Let Y_{ff}, Y_{ft}, Y_{tf} and Y_{tt} be $n_l \times 1$ vectors where the lth element of each corresponds to y_{ff}^l, y_{ft}^l, y_{tf}^l and y_{tt}^l, respectively. Furthermore, let C_f and C_t be the $n_l \times n_b$ branch-bus connection matrices, where $C_{fij} = 1$ and $C_{tik} = 1$ if branch i connects from bus j to bus k (Zimmerman, 2010, p.12). The $n_l \times n_b$ branch admittance matrices are

$$
Y_f = \mathbf{diag}(Y_{ff})C_f + \mathbf{diag}(Y_{ft})C_t
\tag{2.16}
$$

$$
Y_t = \mathbf{diag}(Y_{tf})C_f + \mathbf{diag}(Y_{tt})C_t
\tag{2.17}
$$

15

and the $n_b \times n_b$ nodal admittance matrix is

$$Y_{bus} \;\; = \;\; C_f^{\mathsf{T}} Y_f + C_t^{\mathsf{T}} Y_t. \tag{2.18}$$

Power Balance   For a network of n_b nodes, the current injected at node i is

$$I_i = \sum_{j=1}^{n_b} Y_{ij} V_j \tag{2.19}$$

where $Y_{ij} = |Y_{ij}| \angle \theta_{ij}$ is the $(i,j)^{th}$ element if the Y_{bus} matrix. Hence, the apparent power entering the network at bus i is

$$S_i = P_i + Q_i = V_i I_i^* = \sum_{n=1}^{n_b} |Y_{ij} V_i V_j| \angle (\delta_i - \delta_j - \theta_{ij}) \tag{2.20}$$

Converting to polar coordinates and separating the real and imaginary parts, the active power

$$P_i = \sum_{n=1}^{n_b} |Y_{ij} V_i V_j| \cos(\delta_i - \delta_j - \theta_{ij}) \tag{2.21}$$

and the reactive power

$$Q_i = \sum_{n=1}^{n_b} |Y_{ij} V_i V_j| \sin(\delta_i - \delta_j - \theta_{ij}) \tag{2.22}$$

entering the network at bus i are non-linear functions of V_i, as indicated by the presence of the sine and cosine terms. Kirchhoff's Current Law requires that the net complex power injection (generation - load) at each bus equals the sum of complex power flows on each branch connected to the bus. The power balance equations

$$P_g^i - P_d^i = P^i \tag{2.23}$$

and

$$Q_g^i - Q_d^i = Q^i, \tag{2.24}$$

where the subscripts g and d indicate generation and demand respectively, form the principal non-linear constraints in the optimal power flow problem.

Optimal Power Flow Formulation

Optimal power flow is a mathematical optimisation problem constrained by the complex power balance equations (2.23) and (2.24). Mathematical optimisation

problems have the general form

$$\min_{x} f(x) \qquad (2.25)$$

subject to

$$g(x) = 0 \qquad (2.26)$$
$$h(x) \leq 0 \qquad (2.27)$$

where x is the vector of optimisation variables, f is the objective function and equations (2.26) and (2.27) are sets of equality and inequality constraints, respectively, on x.

In optimal power flow, typical inequality constraints are bus voltage magnitude contingency state limits, generator output limits and branch power or current flow limits. The vector of optimisation variables x may consist of generator set-points, bus voltages, transformer tap settings etc. If x is empty then the formulation reduces to the general power flow problem described above.

A common objective in the optimal power flow problem is total system cost minimisation. For a network of n_g generators the objective function is

$$\min_{\theta, V_m, P_g, Q_g} \sum_{k=1}^{n_g} c_P^k(p_g^k) + c_Q^k(q_g^k) \qquad (2.28)$$

where c_P^k and c_Q^k are cost functions (typically quadratic) of the set-points p_g^k and q_g^k for generator k, respectively. Alternative objectives may be to minimise losses, maximise the voltage stability margin or minimise deviation of an optimisation variable from a particular schedule (Kallrath, Pardalos, Rebennack, & Scheidt, 2009, §18).

Nodal Marginal Prices

Many solution methods for optimal power flow have been developed since the problem was introduced by Carpentier (1962) and a review of the main techniques can be found in Momoh, Adapa, and El-Hawary (1999); Momoh, El-Hawary, and Adapa (1999). One of the most robust strategies is to solve the Lagrangian function

$$\mathcal{L}(x) = f(x) + \lambda^{\mathsf{T}} g(x) + \mu^{\mathsf{T}} h(x), \qquad (2.29)$$

where λ and μ are vectors of Lagrangian multipliers, using an Interior Point Method (Boyd & Vandenberghe, 2004). When solved, the Lagrangian multiplier

for a constraint gives the rate of change of the objective function value with respect to the constraint variable. If the objective function is equation (2.28), the Lagrangian multipliers λ_P^i and λ_Q^i for the power balance constraint at each bus i, given by equations (2.23) and (2.24), are the nodal marginal prices and can be interpreted as the increase in the total system cost for and additional injection at i of 1MW or 1MVAr, respectively.

For a case in which none of the inequality constraints $h(x)$ (such as branch power flow or bus voltage limits) are binding, the nodal marginal prices are uniform across all buses and equal the cost of the marginal generating unit. When the constraints *are* binding, the nodal marginal prices are elevated for buses at which adjustments to power injection are required for the constraints to be satisfied. Nodal marginal prices are commonly used in agent-based electricity market simulation to determine the revenue for generating units as they reflect the increased value of production in constrained areas of the power system.

2.4   Reinforcement Learning

Reinforcement learning is learning from reward by mapping situations to actions when interacting with an uncertain environment (Sutton & Barto, 1998). An agent learns *what* to do in order to achieve a task through trial-and-error using a numerical reward or a penalty signal without being instructed *how* to achieve it. Some actions may not yield immediate reward or may effect the next situation and all subsequent rewards. Always, a compromise must be made between the exploitation of past experiences and the exploration of the environment through new action choices. In reinforcement learning an agent must be able to:

- Sense aspects of its environment,

- Take actions that influence its environment and,

- Have an explicit goal or set of goals relating to the state of its environment.

In the classical model of agent-environment interaction, at each time step t in a sequence of discrete time steps $t = 1, 2, 3 \ldots$ an agent receives as input some form of the environment's state $s_t \in S$, where S is the set of possible states. From a set of actions $A(s_t)$ available to the agent in state s_t and the agent selects an action a_t and performs it in its environment. The environment enters a new state s_{t+1} in the next time step and the agent receives a scalar numerical reward $r_{t+1} \in \mathbb{R}$ in part as a result of its action. The agent then learns from the state representation, the chosen action a_t and the reinforcement signal r_{t+1} before beginning its next interaction. Figure 2.5 diagrams the classical agent-environment interaction event sequence in reinforcement learning.

For a finite number of states, if all states are Markov, the agent interacts with a finite Markov decision process (MDP). Informally, for a state to be Markov it must retain all relevant information about the complete sequence of positions leading up to the state, such that all future states and expected rewards can be predicted as well as would be possible given a complete history (Sutton & Barto, 1998). A particular MDP is defined for a discrete set of time steps by a state set S, an action set A, a set of state transition probabilities P and a set of expected reward values R. In practice not all state signals are Markov, but should provide a good basis for predicting subsequent states, future rewards and selecting actions.

If the state transition probabilities and expected reward values are not known, only the states and actions, then samples from the MDP must be taken and a value function approximated iteratively based on new experiences generated by performing actions.
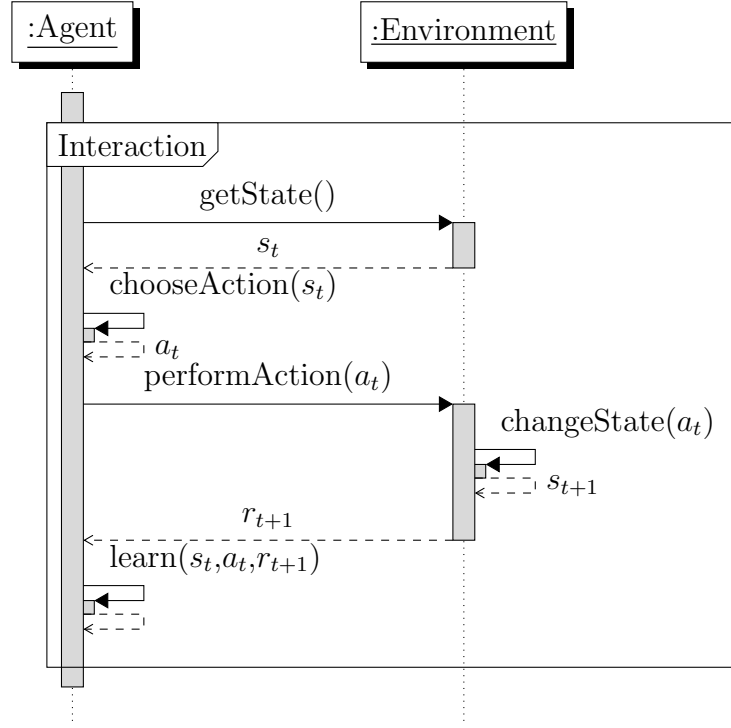
Figure 2.5: Sequence diagram for the basic reinforcement learning model.

2.4.1 Value Function Methods

Any method that can optimise control of a MDP may be considered a reinforcement learning method. All search for an optimal policy π^* that maps state $s \in \mathcal{S}$ and action $a \in \mathcal{A}$ to the probability $\pi^*(s, a)$ of taking a in s and maximises the sum of rewards over the agents lifetime.

Each state s under policy π may be associated with a *value* $V^\pi(s)$ equal to the expected return from following policy π from state s. Most reinforcement learning methods are based on estimating the state-value function

$$V^\pi(s) = E\left\{ \sum_{t=0}^{\infty} \gamma^t r_t \middle| s_0 = s \right\} \tag{2.30}$$

where γ is a discount factor, with $0 \leq \gamma \leq 1$ and E indicates that it is an estimate. Performing certain actions may result in no state change, creating a loop and causing the value of that action to be infinite for certain policies. The discount factor γ prevents values from going unbounded and represents reduced trust in the reward r_t as discrete time t increases. Many reinforcement learning

methods estimate the action-value function

$$Q^\pi(s, a) = E\left\{ \sum_{t=0}^{\infty} \gamma^t r_t \middle| s_0 = s, a_0 = a \right\} \tag{2.31}$$

which defines the value of taking action a in state s under fixed policy π.

Temporal-Difference Learning

Temporal Difference (TD) learning is a fundamental concept in reinforcement learning that was introduced by Sutton and Barto (1998). TD methods do not attempt to estimate the state transition probabilities and expected rewards of the finite MDP, but estimate the value function directly. They learn to *predict* the expected value of total reward returned by the state-value function (2.30). For an exploratory policy π and a non-terminal state s, an estimate of $V^\pi(s_t)$ at any given time step t is updated using the estimate at the next time step $V^\pi(s_{t+1})$ and the observed reward r_{t+1}

$$V^\pi(s_t) = V^\pi(s_t) + \alpha\left[r_{t+1} + \gamma V^\pi(s_{t+1}) - V^\pi(s_t)\right] \tag{2.32}$$

where α is the learning rate, with $0 \leq \alpha \leq 1$, which controls how much attention is paid to new data when updating V^π. Plain TD learning evaluates a particular policy and offers strong convergence guarantees, but does not learn better policies.

Q-Learning

Q-learning is an off-policy TD method that does not estimate the finite MDP directly, but iteratively approximates a state-action value function which returns the value of taking action a in state s and following an *optimal* policy thereafter. The same theorems used in defining the TD error also apply for state-action values that are updated according to

$$Q(s_t, a_t) = Q(s_t, a_t) + \alpha\left[r_{t+1} + \gamma \max_a Q(s_{t+1}, a) - Q(s_t, a_t)\right]. \tag{2.33}$$

The method is off-policy since the update function is independent of the policy being followed and only requires that all state-action pairs be continually updated.

Sarsa

Sarsa (or modified Q-learning) is an on-policy TD control method that approximates the state-action value function in equation (2.31). Recall that the state-

action value function for an agent returns the total expected reward for following a particular policy for selecting actions as a function of future states. The function is updated according to the rule

$$Q(s_t, a_t) = Q(s_t, a_t) + \alpha \big[r_{t+1} + \gamma Q(s_{t+1}, a_{t+1}) - Q(s_t, a_t) \big]. \qquad (2.34)$$

This update also uses the action from the next time step a_{t+1} and the requirement to transition through state-action-reward-state-action for each time step gives the algorithm its name. Sarsa is referred to as an on-policy method since it learns the same policy that it follows.

Eligibility Traces

With the TD methods described above, only the value for the immediately preceding state or state-action pair is updated at each time step. However, the prediction $V(s_{t+1})$ also provides information concerning earlier predictions and TD methods can be extended to update a set of values at each step. An eligibility trace $e(s)$ represents how eligible the state s is to receive credit or blame for the TD error:

$$\delta = r_{t+1} + \gamma V^\pi(s_{t+1}) - V^\pi(s_t) \qquad (2.35)$$

When extended with eligibility traces TD methods update values for all states

$$\Delta V_t(s) = \alpha \delta_t e_t(s) \qquad (2.36)$$

For the current state $e(s) = e(s) + 1$ and for all states $e(s) = \gamma \lambda e(s)$ where λ is the eligibility trace attenuation factor from which the extended TD methods TD(λ), Q(λ) and Sarsa(λ) derive their names. For $\lambda = 0$ only the preceding value is updated, as in the unextended definitions, and for $\lambda = 1$ all preceding state-values or state-action values are updated equally.

Action Selection

A balance between exploration of the environment and exploitation of past experience must be struck when selecting actions. The ϵ-greedy approach to action selection is defined by a randomness parameter ϵ and a decay parameter d. A random number x_r where $0 \leq x_r \leq 1$ is drawn for each selection. If $x_r < \epsilon$ then a random action is selected, otherwise the perceived optimal action is chosen. After each selection the randomness is attenuated by d.

Action selection may also be accomplished using a form of the *softmax* method

(Sutton & Barto, 1998, §2) using the Gibbs (or Boltzmann) distribution to select action k for the $(t+1)^{th}$ interaction with probability

$$p_{jk}(t+1) = \frac{e^{q_{jk}(t+1)/\tau}}{\sum_{l=0}^{K} e^{q_{jl}(t+1)/\tau}} \tag{2.37}$$

where τ is the *temperature* parameter. This parameter may be lowered in value over the course of an experiment since high values give all actions similar probability and encourage exploration of the action space, while low values promote exploitation of past experience.

2.4.2   Policy Gradient Methods

Value function based methods have been successfully applied with discrete look-up table parameterisation to many problems (Bertsekas & Tsitsiklis, 1996). However, the number of discrete states required increases rapidly as the dimensions of the state space increase and if all possibly relevant situations are to be covered, these methods become subject to Bellman's Curse of Dimensionality (Bellman, 1961). Value function based methods can be used in conjunction with function approximation techniques (artificial neural networks typically) to allow operation with continuous state and action spaces. However, greedy action selection has been shown to cause these methods to exhibit poor convergence or divergence characteristics, even in simple systems (Tsitsiklis & Roy, 1994; Peters & Schaal, 2008; Gordon, 1995; Baird, 1995).

These convergence problems have motivated research into alternative learning methods (such as policy gradient algorithms) that can operate with function approximators. Policy gradient algorithms make small incremental changes to the parameter vector θ of a policy function approximator. Using artificial neural networks the parameters are the weights of the network connections. Policy gradient methods update θ in the direction of steepest ascent of some policy performance measure Y with respect to the parameters

$$\theta_{i+1} = \theta_i + \alpha \frac{\partial Y}{\partial \theta_i} \tag{2.38}$$

where α is a positive definite step size learning rate. Unlike look-up table based methods, they do not require all states to be continually updated. Uncertainty in state data can degrade policy performance, but the methods generally have strong convergence properties.

Policy gradient methods are differentiated largely by the techniques used to
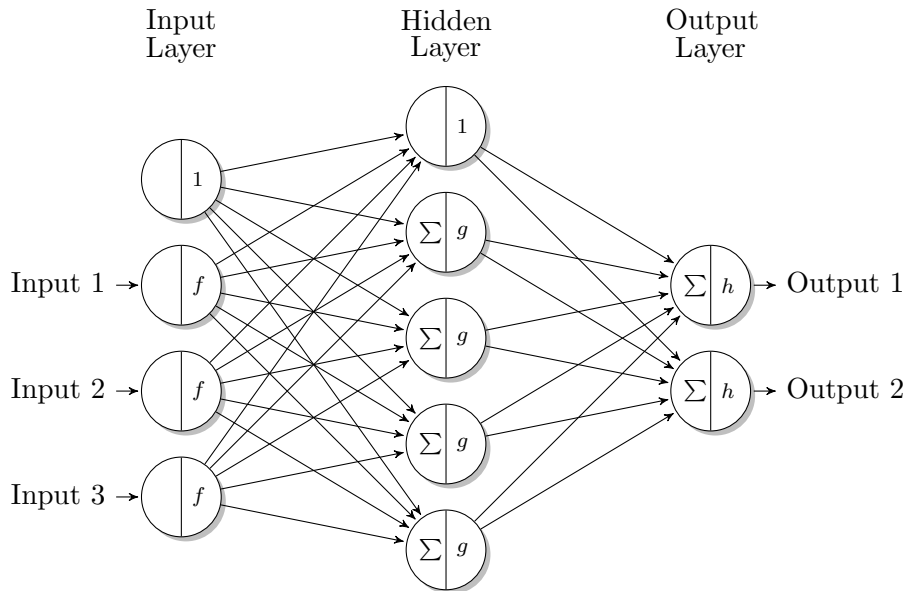
Figure 2.6: Multi-layer feed-forward perceptron with bias nodes.

obtain an estimate of the policy gradient $\partial Y / \partial \theta$. Some of the most successful real-world robotics results (Glynn, 1987; Aleksandrov, Sysoyev, & Shemeneva, 1968) have been yielded by likelihood ratio methods such as Williams' REINFORCE (Williams, 1992) and natural policy gradient methods, such as the Episodic Natural Actor-Critic (ENAC) (Peters & Schaal, 2008). These algorithms have lengthy derivations, but an overview has been published by Peters (2010).

Artificial Neural Networks

Artificial neural networks are mathematical models that mimic aspects of biological neural networks, such as the human brain, and are widely used in supervised learning applications (Bishop, 1996; Fausett, 1994). In reinforcement learning, the most widely used type of artificial neural network is the multi-layer feed-forward network (or multi-layer perceptron). This model consists of an input layer and an output layer of artificial neurons, plus any number of optional hidden layers. Weighted connections link neurons, but unlike architectures such as the recurrent neural network, only neurons from adjacent layers are connected. Most commonly, a fully connected scheme is used in which all neurons from one layer are connected to all neurons in the next. Figure 2.6 diagrams a fully connected three layer feed-forward neural network.

McCulloch and Pitts (1943) conceived of an artificial neuron j that computes

24

a function g as a weighted sum of all n inputs

$$y_j(x) = g\left(\sum_{i=0}^{n} w_i x_i\right) \tag{2.39}$$

where $(w_0 \ldots w_n)$ are weights applied to the inputs $(x_0 \ldots x_n)$. In an multi-layer neural network the output y_j forms part of the input to the neurons in any following layer. The activation function g is typically either:

- Linear, where $y_j = \sum_{i=0}^{n} w_i x_i$,

- A threshold function, with $y_j \in \{0, 1\}$,

- Sigmoidal, where $0 \le y_j \le 1$, or

- A hyperbolic tangent function, where $-1 \le y_j \le 1$.

The parameters of the activation functions can be adjusted along with the connection weights to tune the transfer function between input and output that the network provides. To simplify this process a *bias* node that always outputs 1 may be added to a layer and connected to all neurons in the following layer. This can be shown to allow the activation function parameters to be removed and for network adjustment to be achieved using only connection weights.

The output is obtained during the network's *execution* phase by presenting an input to the input layer that propagates through. It can be shown that a suitably configured feed-forward network with one hidden layer can approximate any non-linear function.

2.4.3 Roth-Erev Method

The reinforcement learning method formulated by Alvin E. Roth and Ido Erev is based on empirical results obtained from observing how humans learn decision making strategies in games against multiple strategic players (Roth et al., 1995; Erev & Roth, 1998). It learns a stateless policy in which each action a is associated with a value q for the propensity of its selection. In time period t, if agent j performs action a' and receives a reward $r_{ja'}(t)$ then the propensity value for action a at time $t + 1$ is

$$q_{ja}(t+1) = \begin{cases} (1-\phi)q_{ia}(t) + r_{ja'}(t)(1-\epsilon), & a = a' \\ (1-\phi)q_{ia}(t) + r_{ja'}(t)(\frac{\epsilon}{A-1}), & a \ne a' \end{cases} \tag{2.40}$$

where A is the total number of feasible actions, ϕ is the *recency* parameter and ϵ is the *experimentation* parameter. The recency (forgetting) parameter degrades the propensities for all actions and prevents propensity values from going unbounded. It is intended to represent the tendency for players to forget older action choices and to prioritise more recent experience. The experimentation parameter prevents the probability of choosing an action from going to zero and encourages exploration of the action space.

Erev and Roth proposed action selection according to a discrete probability distribution function, where action k is selected for interaction $t+1$ with probability

$$p_{jk}(t+1) = \frac{q_{jk}(t+1)}{\sum_{l=0}^{K} q_{jl}(t+1)} \qquad (2.41)$$

Since $\sum_{l=0}^{K} q_{jl}(t+1)$ increases with t, a reward $r_{jk}(t)$ for performing action k will have a greater effect on the probability $p_{jk}(t+1)$ during early interactions while t is small. This is intended to represent Psychology's Power Law of Practice in which it is qualitatively stated that with practice learning occurs at a decaying exponential rate and that a learning curve will eventually flatten out.

Modified Roth-Erev Method

Two shortcomings of the basic Roth-Erev algorithm have been identified and a modified formulation proposed by Nicolaisen, Petrov, and Tesfatsion (2002). The two issues are that

- the values by which propensities are updated can be zero or very small for certain combinations of the experimentation parameter ϵ and the total number of feasible actions A and

- all propensity values are decreased by the same amount when the reward, $r_{jk'}(t)$ is zero.

Under the variant algorithm, the propensity for agent j to select action a for interaction $t+1$ is:

$$q_{ja}(t+1) = \begin{cases} (1-\phi)q_{ia}(t) + r_{ja'}(t)(1-\epsilon), & a = a' \\ (1-\phi)q_{ia}(t) + q_{ja}(t)(\frac{\epsilon}{A-1}), & a \neq a' \end{cases} \qquad (2.42)$$

As with the original Roth-Erev algorithm, the propensity for selection of the action that the reward is associated with is adjusted by the experimentation

parameter. All other action propensities are adjusted by a small proportion of their current value.

Stateful Roth-Erev

The Roth-Erev technique maintains a single vector of propensities for each action. Action-value function based methods, such as Q-learning and Sarsa, typically update a matrix, or look-up table, where each row corresponds to an individual state. In this thesis a *Stateful Roth-Erev* method is proposed. The method is a simple extension to the original or modified version that maintains an action propensity *matrix* with a row corresponding to each discrete state. Updates are done according to equation (2.40) or equation (2.42), but only action propensities for the current state are updated. The method allows for differentiation between states of the environment, but can greatly increase the number of propensity values requiring updates.

2.5   Summary

The combination of an electricity market and an electric power system presents a complex dynamic environment for participants. Network power flows are non-linear functions of the bus voltages and thus one party's generation or consumption decisions effect all other parties.

The main electricity trading mechanisms can be modelled using well established mathematical optimisation formulations. Robust techniques exist for computing solutions to these problems, which also provide price information that reflects the network topology and conditions. The combination of non-linear optimisation problems and participant behavioural models is beyond the capabilities of conventional equilibrium approaches to market simulation when analysing large systems. An alternative is to take a "bottom-up" modelling approach to them and examine the system dynamics that result from interactions between goal driven individuals.

Reinforcement learning is an unsupervised machine learning technique that can be used to model the dynamic behaviour of these individuals. Traditional methods associated a *value* with each state and the available actions, but are limited to small discrete problem representations. Policy gradient methods that search directly in the space of the parameters of an action selection policy can operate in continuous environments, have been shown to exhibit good convergence

properties and have been successfully applied in laboratory and operational settings.

Chapter 3

Related Work

This chapter describes the research in this thesis in the context of similar work. It focuses on the learning methods and simulation models used in previously published research. For a similar review with greater emphasis on simulation results and conclusions drawn from them, the interested reader is referred to Weidlich and Veit (2008).

3.1 Custom Learning Methods

The earliest agent-based electricity market simulations in the literature do not use traditional learning methods from the field of Artificial Intelligence, but rely upon custom heuristic methods. These are typically formulated using the author's intuition and represent basic trading rules, but do not encapsulate many of the key concepts from reinforcement learning theory.

3.1.1 Market Power

Under Professor Derek Bunn, researchers from the London Business School performed some the first and most reputable agent-based electricity market simulations. Their research was initially motivated by proposals in 1999 to transform the structure of The England and Wales Electricity Pool, with the aim of combating the perceived generator market power that was believed to be resulting in elevated market prices.

In Bower and Bunn (2001) a detailed model of electricity trading in England and Wales is used to compare day-ahead and bilateral contract markets under uniform price and discriminatory settlement. Twenty generating companies operating in the Pool during 1998 are modelled as agents endowed with portfolios

of generating plant. Plant capacities, costs and expected availabilities are synthesised from public and private data sources and the author's own estimates. In simulations of the day-ahead market, each agent submits a single price for the following simulated trading day, for each item of plant in its portfolio. Whereas, under the bilateral contract model, 24 bids are submitted for each generator, corresponding to each hour of the following simulated day. Revenues are calculated at the end of each trading day and are determined either by the bid price of the marginal unit or the generator's own bid price. Each generating plant is characterised in part by an estimated target utilisation rate that represents its desire for forward contract cover. The agents learn to achieve this utilisation rate and then to improve profitability.

If the utilisation rate is not achieved, a random percentage from a uniform distribution with a range of $\pm10\%$ and 0% mean is subtracted from the bid price of all generators in the agent's portfolio. Agents with more than one generator transfer successful bidding strategies between plant by setting the bid price for a generator to the level of the next highest submitted bid price if the generator sold at a price lower than that of other generators in the same portfolio. If an agent's total profit does not increase, a random percentage from the same distribution as above is added or subtracted from the bid price from the previous day for each of its generators. A cap on bid prices is imposed at £1000 in each period. Demand follows a 24-hour profile based on the 1997-1998 peak winter load pattern. The response of the load schedule to high prices is modelled as a reduction of 25MW for every £1/MWh that the system marginal price rises above £75/MWh.

In total, 750 trading days are simulated for each of the four combinations of a day-ahead market and the bilateral trading model under uniform pricing and discriminatory settlement. Prices are found to generally be higher under pay-as-bid pricing for both market models. Agents with larger portfolios are shown to have a significant advantage over smaller generators due to their greater ability to gather scarce market price information and distribute it among generators.

The existence of market power is a common research question in agent-based electricity market simulation and the paper uses a relatively simple learning method to try to answer it. This is a good example of how such simulations need not be restricted to simple models, but can be used to study systems on a national scale.

In Bower, Bunn, and Wattendrup (2001) a more sophisticated custom learning method, resembling the Roth-Erev method, is applied to a detailed model of the New Electricity Trading Arrangements. The balancing mechanism is modelled

as a one-shot market, that follows the contracts market, to which increment and decrement bids are submitted. Active demand side participation is modelled and generator dynamic constraints are represented by limiting the number of off/on cycles per day. Again, transmission constraints and regional price variations are ignored.

Supplier and generator agents are assigned an optimal value for exposure to the balancing mechanism that is set low due to high price and volume uncertainty. The agents learn to maximise profit, but profits are penalised if the objective for balancing mechanism exposure is not achieved. They learn policies for pricing markups on the bids submitted to the power exchange and the increments and decrements submitted to the balancing mechanism. Markups in the power exchange are relative to prices from the previous day and markups on balancing mechanism bids are relative to power exchange bid prices on the same day. Different markup ranges are specified for generators and suppliers in the power exchange and balancing mechanism and each is partitioned into ten discrete intervals.

As with the Roth-Erev method, a probability for the selection of each markup is calculated by the learning method. Daily profits and acceptance rates for bids/offers from previous trading days are extrapolated out to determine expected values and thus the expected reward for each markup. The markups are then sorted according to expected reward in descending order. The perceived utility of each markup j is

$$U_j = \mu \left(\frac{\phi - n}{\phi} \right)^{i_j - 1} \tag{3.1}$$

where i is the index of j in the ordered vector of markups and ϕ is a search parameter. High values of ϕ cause the agent to adopt a more exploratory markup selection policy. For all of the experiments $\mu = 1000$, $\phi = 4$, $n = 3$ and the probability of selecting markup j is

$$Pr_j = \frac{U_j}{\sum_{k=1}^{K} U_k} \tag{3.2}$$

for K possible markups.

A representative model of the England and Wales system with 24 generator agents, associated with a total of 80 generating units, and 13 supplier agents is analysed over 200 simulated trading days. The authors draw conclusions on the importance of accurate forecasts, greater risk for suppliers than generators, the value of flexible plant and the influence of capacity margin on opportunities

for collusive behaviour. The same learning method is applied in D. W. Bunn and Oliveira (2003) as part of an inquiry by the Competition Commission into whether two specific companies in the England and Wales electricity market had enough market power to operate against the public interest.

These papers show a progression towards more complex participant and market models. The work neglects all transmission system constraints, but is an ambitious attempt to relate results to consequences for a national market.

Visudhiphan and Ilic (1999) is another early publication on agent-based simulation of electricity markets in which a custom learning method is used. The simulations comprise only three generators, market power is assumed, and the authors analyse the mechanisms by which the market power is exercised. Two bid formats are modelled. The single-step supply function (SSF) model requires each generator agent to submit a price and a quantity, where the quantity is determined by the generator's marginal cost function. The linear supply function (LSF) model requires each generator agent to submit a value corresponding to the slope of the function. The bid price or slope value for generator i after simulation period t is

$$x_i(t+1) = x_i(t) + b_i(p_m(t))u_i(t) \tag{3.3}$$

where $b_i \in \{-1, 0, 1\}$ is the reward as a function of the market clearing price p_m from stage t and u_i is a reward gain or attenuation parameter. The calculation of b_i is defined according to strategies for estimated profit maximisation and competition to be the base load generator. Both elastic and inelastic load models are considered. Using the SSF model, the two strategies are compared in a day-ahead market setting, using a case where there is sufficient capacity to meet demand and a case where there is excessive capacity to the point where demand can be met by just two of the generators. The LSF model is analysed using both day-ahead and hour-ahead markets with inelastic load. The hour-ahead simulation is repeated with elastic demand response.

The number of if-then rules required to define participant strategies in this paper is demonstrates a drawback of implementing custom learning methods that is only exacerbated when defining multiple strategies.

A similar custom learning method is compared with two other algorithms in Visudhiphan (2003). The custom method is designed specifically for the power pool model that is used and employs separate policies for selecting bid quantities and prices according to several if-then rules that attempt to capture capacity withholding behaviour. The method is compared with algorithms developed in Auer, Cesa-Bianchi, Freund, and Schapire (2003) for application to the n-armed

bandit problem (Robbins, 1952; Sutton & Barto, 1998, §2.1) and a method based on evaluative feedback with softmax action selection.

In the algorithms from Auer et al. (2003) each action $i = 1, 2, \ldots K$, for K possible actions, is associated with a weight $w_t(i)$ in simulation period $t \in T$, for T simulation periods, that is used in determining the action's probability of selection

$$p_i(t) = (1 - \gamma) \frac{w_i(t)}{\sum_{j=1}^{K} w_j(t)} + \frac{\gamma}{K} \tag{3.4}$$

where γ is a tuning parameter, with $0 < \gamma \le 1$, that is initialised such that

$$\gamma = \min\left\{\frac{3}{5}, 2\sqrt{\frac{3}{5} \frac{K \ln K}{T}}\right\}. \tag{3.5}$$

Using the received reward $x_t(i_t)$, the weight for action j in period $t + 1$ is

$$w_{t+1}(j) = w_t(i) \exp\left(\frac{\gamma}{3K}\left(\hat{x}_t(i) + \frac{\alpha}{p_t(i)\sqrt{KT}}\right)\right) \tag{3.6}$$

where

$$\hat{x}_t(i) = \begin{cases} x_t(j)/p_t(i) & \text{if } j = i_t \\ 0 & \text{otherwise} \end{cases} \tag{3.7}$$

and

$$\alpha = 2\sqrt{\ln(KT/\gamma)}. \tag{3.8}$$

In the evaluative feedback method from Sutton and Barto (1998, §2) each action i has a value $Q_t(i)$ in simulation period t equal to the expected average reward if that action is selected. The value of action i in the $(t + 1)^{th}$ period is

$$Q_{t+1}(i) = \begin{cases} (1 - \alpha)Q_t(i) + \alpha r_t(i) & \text{if } i_{t+1} = i \\ Q_t(i) & \text{otherwise} \end{cases} \tag{3.9}$$

where α is a constant *step-size* parameter with $0 < \alpha \le 1$.

Extensive simulation results are presented and the choice of learning method is found to have a significant impact on agent performance, but no quantitative comparison measure is provided and no conclusions are drawn as to which method is superior.

3.1.2   Financial Transmission Rights

In Ernst, Minoia, and Ilic (2004) a custom learning method is defined and used to study generator and supplier profits where financial transmission rights are included in the electricity market. A two node transmission system is defined with one lossless transmission line of limited capacity that is endowed to a transmission operator agent. Generator agents submit bids for their respective generating units and the transmission owner submits a bid representing the cost per MW of transmitting power between the nodes. The market operator clears the bids, minimising costs while balancing supply and demand and not breaching the capacity of the line. Prices at each node are calculated to provide a signal to the agents that captures both energy and transmission costs.

Each agent selects its bid according to a calculation of the reward that it would expect to receive if all other agents were to bid as they did in the previous stage. If multiple bids are found to have the same value then the least expensive is selected. In the first period, previous bids are assumed to be at marginal cost.

Several case studies are examined, with different numbers of generators and line capacities, but few explicit conclusions are drawn. Financial transmission rights are an important issue in electricity markets, but the learning algorithm and network model are perhaps overly simple for practical conclusions to be drawn.

3.2   Simulations Applying Q-learning

More recent agent-based simulations of electricity markets has been carried out with participant's behavioral aspects modelled using Q-learning methods.

3.2.1   Nash Equilibrium Convergence

The most prominent work in which Q-learning is applied comes from the Swiss Federal Institutes of Technology in Zurich and Lausanne. The foundations for this work are laid in Krause et al. (2004) with a comparison of agent-based modelling using reinforcement learning and Nash equilibrium analysis when assessing network constrained power pool market dynamics. Parameter sensitivity of the comparison results is analysed in Krause et al. (2006).

The authors model a mandatory spot market which is cleared using a DC optimal power flow formulation. A five bus power system model is defined with three generators and four inelastic and constant loads. Linear marginal cost

functions

$$c_{g,i}(p_{g,i}) = b_{g,i} + s_{g,i}p_{g,i} \tag{3.10}$$

are defined for each generator i where $p_{g,i}$ is the active power output, $s_{g,i}$ is the slope of the cost function and $b_{g,i}$ is the intercept. Suppliers are given the option to markup their bids to the market, not by increasing $s_{g,i}$, but by increasing $b_{g,i}$ by either 0, 10, 20 or 30%.

Nash equilibrium is computed by clearing the market for all possible markup combinations and determining the actions for which no player is motivated to deviate from, as it would result in a decrease in expected reward. Experiments are conducted in which there is a single Nash equilibrium and where there are two Nash equilibria.

An ϵ-greedy strategy (Sutton & Barto, 1998) is applied for action selection and a *stateless* action value function is updated at each time step t according to

$$Q(a_t) = Q(a_t) + \alpha(r_{t+1} - Q(a_t)) \tag{3.11}$$

where α is the learning rate. Further to Krause et al. (2004), simulations with discrete sets of values for the parameters α and ϵ were carried out in Krause et al. (2006). While parameter variations effected the frequency of equilibrium oscillations, Nash equilibrium was still approached and the oscillatory behaviour observed for almost all of the combinations.

The significance of this research is the verification that the agent-based approach settles at the same theoretical optimum as with closed-form equilibrium approaches and that exploratory policies result in the exploitation of multiple equilibria if they exist.

Convergence to a Nash equilibrium is also shown in Naghibi-Sistani, Akbarzadeh-Tootoonchi, Javidi-D.B., and Rajabi-Mashhadi (2006). Boltzmann (soft-max) exploration is used for action selection with the temperature parameter adjusted during the simulations. A modified version of the IEEE 30 bus test system is used with the number of generators reduced from nine to six. No optimal power flow formulation or details of the reward signal used are provided. Generators are given a three step action space where the slope of a linear supply function may be less than, equal to or above marginal cost. The experimental results show that, with temperature parameter adjustment, Nash equilibrium is achieved and the oscillations associated with ϵ-greedy action selection are avoided.

3.2.2 Congestion Management Techniques

Having validated the suitability of an agent-based, bottom-up, approach to assessing the evolution of market characteristics, the authors apply the same technique to compare congestion management schemes in Krause and Andersson (2006). The first scheme considered is locational marginal pricing (or nodal pricing) where congestion is managed by optimising the output of generators with respect to maximum social welfare. The "market splitting" scheme they considered is similar to locational marginal pricing, but the system is subdivided into zones, within which the nodal prices are uniform. The final "flow based market coupling" scheme also features uniform zonal pricing, but uses a simplified representation of the network. Power flows within the zones are not represented and all lines between zones are aggregated into one equivalent interconnector.

As an alternative to the conventional DC optimal power flow formulation, line power flow computation is done using a power transfer distribution factor (PTDF) matrix. The $(i, j)^{th}$ element of the PTDF matrix corresponds to the change in active power flow on line j given an additional injection of 1MW at the slack bus and corresponding withdrawal of 1MW at node i (Grainger & Stevenson, 1994).

The congestion management schemes get evaluated under perfect competition, where suppliers bid at marginal cost, and under oligopolistic competition, in which markups of 5% and 10% can be added to marginal cost. The benefits obtained between reward at marginal cost and a maximum markup are used to assess market power. The experimental results show that market power allocations are different under each of the three constraint management schemes.

This is a compelling example of how optimal power flow can be used with traditional reinforcement learning methods to address an important research question. The decision not to define environment states is unusual for a Q-learning application and the impact of this deserves investigation.

3.2.3 Gas-Electricity Market Integration

The Q-learning method from Krause et al. (2004, 2006) is used to analyse strategic behaviour in integrated electricity and gas markets in Kienzle, Krause, Egli, Geidl, and Andersson (2007). Again, power flows are computed using a PTDF matrix. Pipeline losses in the gas network are approximated using using a cubic function of flow and three combined gas and electricity models are compared.

In the first model, operators of gas-fired power plant submit separate bid functions for gas and electricity. Bids are then cleared as a single optimisation

36

problem. In model two, operators submit one offer for their capacity to convert gas to electricity. In the third model, bids are submitted only to the electricity market, after which gas is purchased regardless of price. Gas supply offers are modelled as a linear function with no strategic involvement. The models are compared in terms of social welfare, using a three bus power system model with three non-gas-fired power plants and one gas-fired plant.

The experimental results show little difference between electricity prices and social welfare prices between the models. However, this research illustrates the interest in and complexity associated with modelling relationships between multiple markets. The authors recognise the need for further and more detailed simulation in order to improve evaluation of market coupling models.

While this work is of a preliminary nature, it is an important step towards achieving greater understanding of interrelationships between gas and electricity markets using agent-based simulation. Further neglect of state information in the Q-learning method possibly alludes to the difficulty of creating discrete representations of largely continuous environments.

3.2.4 Electricity-Emissions Market Interactions

Researchers at the Argonne National Laboratory have published results from a preliminary study of interactions between emissions allowance markets and electricity markets (J. Wang, Koritarov, & Kim, 2009). A cap-and-trade system for emissions is modelled where generator companies are allocated with CO_2 allowances that may subsequently be traded. Generator companies are assumed to have negligible influence on market clearing prices in the emissions market and allowance prices from the European Energy Exchange are used. In the electricity market, an oligopoly structure is assumed and bids are cleared using a DC optimal power flow formulation.

To improve selection of the ϵ parameter for exploratory action selection, a simulated annealing (SA) Q-learning method based on the Metropolis criterion (Guo, Liu, & Malec, 2004) is used. Under this method ϵ is changed at each simulation step to allow solutions to escape from local optima. A two bus system is used to study cases in which: allowance trading is not used, allowances can be exchanged in the emissions market and with variations in the allowance allocations. A one year, hourly load profile with a summer peak is used to model changes in demand. The electricity market is cleared for each simulated hour and the emissions market gets cleared at the end of each simulated week.

The agents learn, when they have a deficit of allowances, to borrow future

allowances in the summer when load and allowance prices are high. Conversely, when having a surplus, they learn to sell at this time. In the third case, the authors show the sensitivity of profits to initial allocations and conclude that the experimental results can not be generalised. The authors cite further model validation and agent learning method improvements as necessary further work.

The complexity of the combined electricity and emissions market model illustrates how the search spaces for learning methods can grow dramatically as models are enlarged.

3.2.5 Tacit Collusion

The SA-Q-learning method is also used in Tellidou and Bakirtzis (2007) by researchers from the University of Thessaloniki to study capacity withholding and tacit collusion among electricity market participants. A mandatory spot market is implemented, where bid quantities may be less than net capacity and bid prices may be marked up upon marginal cost by increasing the slope of a linear cost function. Again the market is cleared using a DC optimal power flow formulation and locational marginal prices are used to calculate profits that are used as the reinforcement signal in the learning process. Demand is assumed to be inelastic and transmission system parameters to be constant between simulation periods.

A simple two node power system model, containing two generators, is used in three test cases. In a reference case, each generator bids full capacity at marginal cost. In the second case, generators bid quantities in steps of 10MW and price markups in steps of €2/MWh. In the third case, the same generation capacity is split among eight identical generators to increase the level of competition. The experimental results show that generators learn to withhold capacity and develop tacit collusion strategies to capture congestion profits.

This work is similar to earlier research from other institutions and makes minimal further contribution. It suggests that there is potential to accelerate advancement in this field through increased collaboration and sharing of software source code.

3.3 Simulations Applying Roth-Erev

Roth and Erev's reinforcement learning method has received considerable attention from the agent-based electricity market simulation community.

3.3.1 Market Power

In Nicolaisen et al. (2002) an agent-based model of a wholesale electricity market with both supply and demand side participation is constructed. It is used to study market power and short-run market efficiency under discriminatory pricing through systematic variation of concentration and capacity conditions.

To model the power system, each trader is assigned values of available transmission capability (ATC) with respect to each of the other traders. Offers from buyers and sellers are matched on a merit order basis, with quantities restricted by ATC values. Two issues with the original Roth-Erev method are observed and the modified version defined in Section 2.4.3 is proposed.

A maximum markup (markdown) of \$40/MWh is specified for each seller (buyer). Traders are not able to make negative profits and the feasible price range is divided into 30 offer prices for 1000 auction rounds cases and 100 offer prices for 10000 auction round cases. The parameters of the Roth-Erev method are calibrated using direct search within reasonable ranges. Nine combinations of buyer and seller numbers and total trading capacities are tested using the calibrated parameter values and *best-fit* values determined empirically in Erev and Roth (1998).

The experimental results show that good market efficiency is achieved under all configurations and sensitivity to method parameter changes is low. Levels of market power are found to be strongly predictive and little difference is found between cases in which opportunistic price offers are permitted and when traders are forced to bid at marginal cost. The results are compared with those from Nicolaisen, Smith, Petrov, and Tesfatsion (2000), in which genetic algorithms are used. The authors conclude that the reinforcement learning approach leads to higher market efficiency due their adaption according to *individual* profits.

Genetic algorithms were a popular alternative to reinforcement learning methods in early agent-based electricity market research. This paper compares the two and illustrates some of the reasons that perhaps explain why they have now been largely abandoned in this field. The modified Roth-Erev method proposed in this paper is later used in several other publications.

Further research from Iowa State University, involving the modified Roth-Erev method, has used the AMES wholesale electricity market test bed. A detailed description of AMES is provided in Appendix A.9 below. In Li and Tesfatsion (2009b) it is used to investigate strategic capacity withholding in a wholesale electricity market design proposed by the U.S. Federal Energy Regulatory Commission in April 2003. A five bus power system model with five generators and

three dispatchable loads is defined and capacity withholding is represented by permitting traders to bid lower than true operating capacity and higher than true marginal costs.

Comparing results from a benchmark case, in which true production costs are reported, but higher than marginal cost functions may be reported, with cases in which reported production limits may be less than the true values, the authors find that with sufficient capacity reserve there is no evidence to suggest potential for inducing higher net earnings through capacity withholding in the market design.

AMES was the first agent-based electricity market simulation program to be released as open source, but while there are several publications on the project, papers involving its application are scarce. This shows how the challenge that is faced if such projects are to benefits from the community collaboration that often leads to the success of open source software projects.

3.3.2   Italian Wholesale Electricity Market

Rastegar, Guerci, and Cincotti (2009) from the University of Genoa used the modified Roth-Erev method to study strategic behaviour in the Italian wholesale electricity market. An accurate model of the actual clearing procedure is implemented and a model of the Italian transmission system, including an interconnector to Sicily and zonal subdivision is defined (See Figure 3.1). Within each of the 11 zones, thermal plant is combined according to technology (coal, oil, combined cycle gas, turbo gas and repower) and associated with one of 16 generation companies according to the size of the companies share. The resulting 53 agents are assumed to bid full capacity and may markup bid prices in steps of 5%, with a maximum markup of 300%.

Bids are cleared using a DC optimal power flow formulation with generation capacity constraints and zone interconnector flow limits. Unusually, the flow limits in the model are different depending on the flow direction: requiring a customised optimal power flow formulation. Agents are rewarded according to a uniform national price, computed as a weighted average of zonal prices with respect to zonal load. Using actual hourly load data it is shown that in experiments in which agents *learn* their optimal strategy, historical trends can be replicated in all but certain hours of peak load. The authors state a desire to test different learning methods and perform further empirical validation.
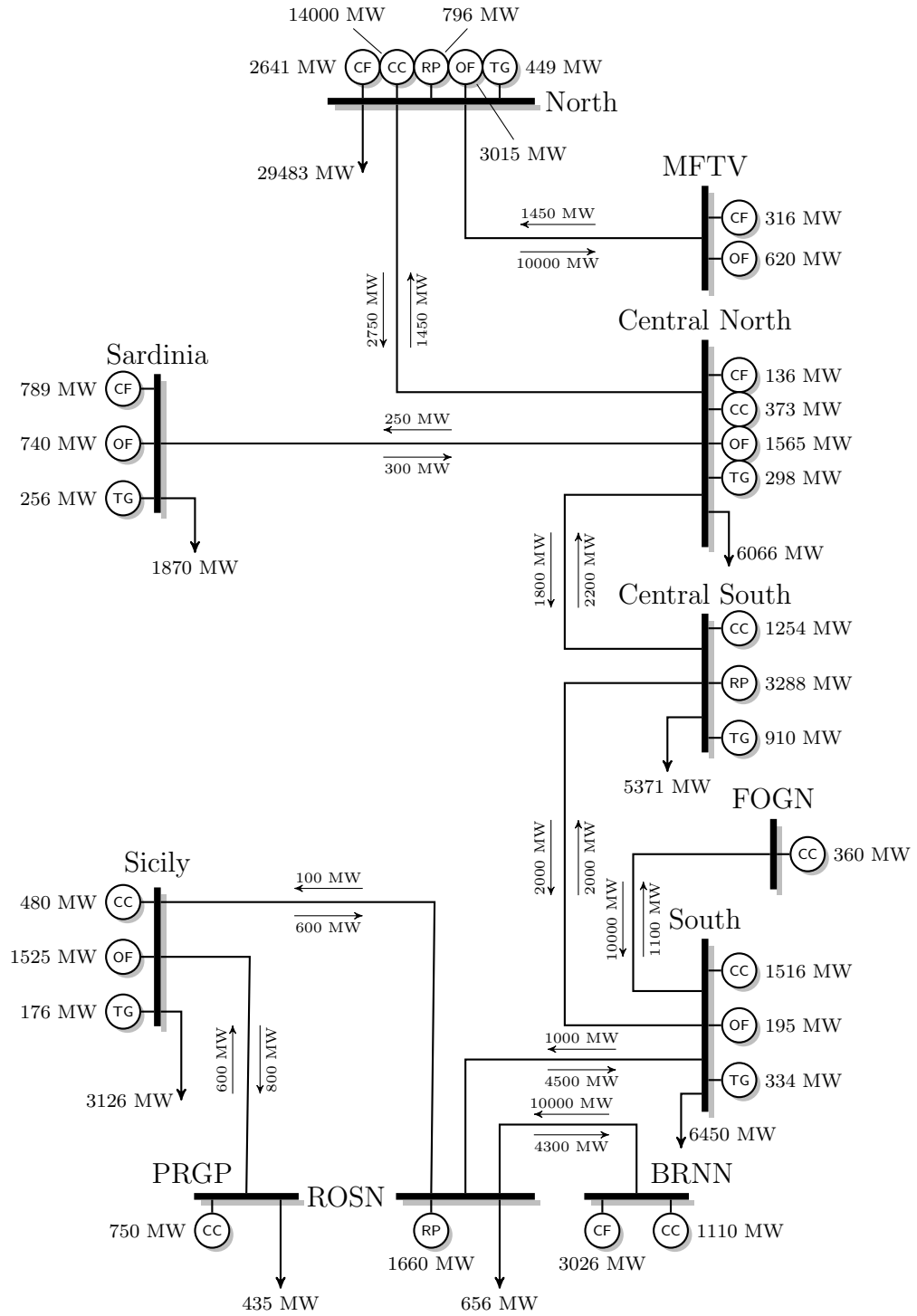
Figure 3.1: Single-line diagram for a stylised Italian grid model.

3.3.3 Vertically Related Firms and Crossholding

In Micola, Banal-Estañol, and Bunn (2008) a multi-tier model of wholesale natural gas, wholesale electricity and retail electricity markets is studied using another variant of the Roth-Erev method. Coordination between strategic business units (SBU) within the same firm, but participating in different markets, is varied systematically and profit differences are analysed.

A two-tier model involving firms with two associated agents whose rewards r_1 and r_2 are initially independent. A "reward independence" parameter α is used to control the fraction of profit from one market that is used in rewarding the agent in the other market. The total rewards are

$$R_1(t) = (1 - \alpha)r_1(t) + \alpha r_2(t) \tag{3.12}$$

and

$$R_2(t) = (1 - \alpha)r_2(t) + \alpha r_1(t). \tag{3.13}$$

Each action a is a single price bid between zero and the clearing price from the preceding market. The Roth-Erev method is modified such that similar actions, $a - 1$ and $a + 1$, are also reinforced. For each agent i, the action selection propensities in auction round t are

$$p_a^i(t) = \begin{cases} (1 - \gamma)p_a^i(t - 1) + R_i(t) & \text{if } s = k \\ (1 - \gamma)p_a^i(t - 1) + (1 - \delta)R_i(t) & \text{if } s = k - 1 \text{ or } s = k + 1 \\ (1 - \gamma)p_a^i(t - 1) & \text{if } s \neq k - 1, s \neq k \text{ or } s \neq k + 1 \end{cases} \tag{3.14}$$

where δ, with $0 \leq \delta \leq 1$, is the local experimentation parameter, γ is the discount parameter and $i \in \{1, 2\}$. Actions whose probability of selection fall below a specified value are removed from the action space.

The initial simulation consists of two wholesalers and three retailers and α is varied from 0 to 0.5 in 51 discrete steps. The experiment is repeated using a three tier model in which two natural gas shippers supply three electricity generators who, in turn, sell to four electricity retailers. The results show a rise in market prices as reward interdependence is increased and greater profits for integrated firms.

The same alternative formulation of the Roth-Erev method is also used in Micola and Bunn (2008) to analyse the effect on market prices of different de-

grees of producer cross-holding[1] under private and public bidding information. Cross-holding is represented with the introduction of a factor to each agent's reward function that controls the fraction of profit from the cross-owned rival that the agent receives. Public information availability is modelled using a vector of probabilities for selection of each possible action that is the average of each agent's private probability and is made available to all agents.

The degree to which the public probabilities influence the agent's action selection probability from equation (2.41) is varied systematically in a series of experiments, along with cross-holding levels and buyer numbers. The results are illustrated using three-dimensional plots and show a direct relationship between cross-holding and market price. The conclusions drawn on market concentration by the authors are dependant upon the ability to model both the demand and supply side participation in the market and the authors state that this shows, to a certain extent, the value of the agent-based simulation approach.

3.3.4 Two-Settlement Markets

In Weidlich and Veit (2006) the modified Roth-Erev method is used to study interrelationships between contracts markets and balancing markets. Bids on the day-ahead contracts market consist of a price and a volume, which are assumed to be the same for each hour of the day. Demand is assumed to be fixed and inelastic. Bids on the balancing market consist of a reserve price, a work price and an offered quantity. The reserve price is that which must be paid for the quantity to be kept on standby and the work price must be paid if that quantity is called upon for transmission system stabilisation. No optimal power flow formulation or power system model is defined.

At the day-ahead stage, contract market and balancing market bids are cleared, according to reserve price, by stacking in order of ascending price until the forecast demand is met. On the following day, accepted balancing bids are cleared according to work price such that requirements for reserve dispatch are met.

Bid prices on the contracts market are stratified into 21 discrete values between 0 and 100 and bid quantities into six discrete values between 0 and maximum capacity, giving 126 possible actions. Bid quantities on the balancing market equal the capacity remaining after contract market participation. 21 discrete capacity prices between 0 and 500 and 5 work prices between 0 and 100 are permitted, giving 105 possible actions in the balancing market. Separate instances

---

[1]Cross-holdings occur when one publicly traded firm owns stock in another such firm.

of the modified Roth-Erev method are used to learn bidding strategies for each agent in each of the markets.

Interrelationships between the markets are studied using four scenarios in which the order of market execution and the balancing market pricing mechanism (discriminatory or pay-as-bid) are changed. Clearing prices in the market executed first are shown to have a marked effect on prices in the following market. The authors find agent-based simulation to be a suitable tool for reproducing realistic market outcomes and recognise a need for more detailed participant models.

In the same year, the authors collaborated with Jian Yao and Shmuel Oren from the University of California to study the dynamics between two settlement markets using the modified Roth-Erev method (Veit, Weidlich, Yao, & Oren, 2006). The markets are a forward contracts market, in which transmission constraints are ignored, and a spot market that is cleared using a DC optimal power flow formulation with line flows calculated using a PTDF matrix. The authors state that suppliers utility functions are to include aspects of risk aversion in future work. The use of some measure of risk adjusted return to assess performance is commonplace in economics research, but is currently lacking from the agent-based electricity market simulation literature.

Zonal prices are set in the forward market as weighted averages of nodal prices with respect to historical load shares. Profits are determined using the zonal prices and nodal prices from optimisation of the spot market. Demand is assumed inelastic to price, but different contingency states with peak and low demand levels are examined. A stylised 53 bus model of the Belgian electricity system from Yao, Oren, and Adler (2007) and Yao, Adler, and Oren (2008) is used to validate the results against those obtained using equilibrium methods. The nineteen generators are divided among two firms which learn strategies for bid price and quantity selection using the modified Roth-Erev method with a set of fixed parameter values taken from Erev and Roth (1998). The results show that the presence of a forward contracts market produces lower overall electricity prices and lower price volatility. The authors note that risk aversion is to be included in suppliers utility functions in future work.

3.4   Policy Gradient Reinforcement Learning

Policy gradient reinforcement learning methods have been successfully applied in both laboratory and operational settings (Sutton, McAllester, Singh, & Mansour, 2000; Peters & Schaal, 2006; Peshkin & Savova, 2002). This section reviews

market related applications of these methods.

3.4.1 Financial Decision Making

Conventionally, *supervised* learning techniques are used in financial decision making problems to minimise errors in price forecasts and are trained on sample data. In Moody, Wu, Liao, and Saffell (1998) a recurrent reinforcement learning method is used to optimise investment performance without price forecasting. The method is "recurrent" in that it uses information from past decisions as input to the decision process. The authors compare direct profit and the Sharpe ratio (Sharpe, 1966, 1994) as reward signals. The Sharpe ratio $S_t = \bar{r}_t/\sigma$ is a measure of risk adjusted return where r_t is the return for period t and σ is the standard deviation.

The parameters θ of the trading system are updated in the direction of the steepest accent of the gradient of some performance function U_t with respect to θ

$$\Delta\theta_t = \rho \frac{dU_t(\theta_t)}{d\theta_t} \tag{3.15}$$

where ρ is the learning rate. Direct profit is the simplest performance function defined, but assumes traders are insensitive to risk. Investors being sensitive to losses are, in general, willing to sacrifice potential gains for reduced risk of loss. To allow on-line learning and parameter updates at each time period, the authors define a *differential* Sharpe ratio. By maintaining an exponential moving average of the Sharpe ratio, the need to compute return averages and standard deviations for the entire trading history at each simulation period is avoided. Alternative performance ratios, including the Information ratio, Appraisal ratio and Sterling ratio, are mentioned.

Simulations are conducted using artificial price data, equivalent to one year of hourly trade in a 24-hour market, and using 45 years of monthly data from the Standard & Poor (S&P) 500 stock index and 3 month Treasury Bill (T-Bill) data. In a portfolio management simulation, in which trading systems invest portions of their wealth among three different securities, it was shown that trading systems maximising the differential Sharpe ratio, produced more consistent results and achieved higher risk adjusted returns than those trained to simply maximise profit. This result is important as the majority of reinforcement learning applications in electricity market simulation use direct profit for the reward signal and may benefit from using measures of risk adjusted return.

In Moody and Saffell (2001) the recurrent reinforcement learning method from

Moody et al. (1998) is contrasted with value function based methods. In addition to the Sharpe ratio, a Downside Deviation ratio is defined. Results from trading systems trained on half-hourly United States Dollar-Great British Pound foreign exchange rate data and again learning switching strategies between the S&P 500 index and T-Bills are presented. They show that the recurrent reinforcement learning method outperforms Q-learning in the S&P 500/T-Bill allocation problem. The authors observe that the recurrent reinforcement learning method has a much simpler functional form in that the output, not being discrete, maps easily to real valued actions and that the algorithm is more robust to noise in the financial data and adapts quickly to non-stationary environments.

3.4.2 Grid Computing

In Vengerov (2008) a marketplace for computational resources in envisioned. The authors propose a market in which grid service suppliers offer to execute jobs submitted by customers for a price per CPU-hour. The problem formulation requires customers to request a quote for computing a job k for a time τ_k on n_k CPUs. The quote returned specifies a price P_k at which k would be charged and a delay time d_k for the job. The service provider's goal is to learn a policy for pricing quotes that maximises long term revenue when competing in a market with other providers. Price differentiation is implemented though provision of a standard service, priced at \$1/CPU-hour and a premium service a \$$P$/CPU-hour, with premium jobs prioritised over standard jobs. The state of the market environment is defined by the current expected delays in the standard and premium service classes and by $n_k \tau_k$: the product of the number of CPUs requested and the job execution time. The reward $r(s, a)$ for action a in state s is the total price paid for the job. The policy gradient method employed is a modified version of REINFORCE (Williams, 1992) where

$$Q(s_t, a_t) = \sum_{t=1}^{T} r(s_t, a_t) - \overline{r}_t \tag{3.16}$$

and $\overline{r}_t$ is the current average reward.

The authors recognise that their grid market model could be generalised to other multi-seller retail markets. The experimental results show that if all grid service providers simultaneously use the learning algorithm then the process converges to a Nash equilibrium. The results also showed that significant increases in profit were possible by offering both standard and premium services.

While this work applies policy gradient methods in a different domain, it shows how these methods can be used to set prices in a market and the author recognises the potential for the approach to be extended to other domains.

3.5 Summary

Agent-based simulation of electricity markets has been a consistently active field of research for more than a decade. Researchers around the world have sought to tackle important Electric Power Engineering problems including:

- Market power,

- Congestion management,

- Tacit collusion,

- Discriminatory vs. pay-as-bid pricing,

- Financial transmission rights, and

- Day ahead markets vs. bilateral trade.

Improvements in these areas have the potential to provide major financial benefits to society.

There is a trend in the literature towards the use of more complex learning methods for participant behavioural representation and increasingly accurate electric power system models. Some of the more ambitious studies have used stylised models of national transmission systems, including the UK, Italy, Belgium and Germany. There have been previous attempts to compare learning methods for simulated electricity trade, but no consensus exists as to which are most appropriate methods for particular applications.

Researchers are extending their studies to investigate energy business structures and the relationships between electricity, fuel and emission allowance markets. Policy gradient reinforcement learning methods have not been previously used in electricity market simulation, but have been shown to work well in similar problems.

Chapter 4

Modelling Power Trade

This chapter defines the model used in chapters 5 and 6 to simulate competitive electric power trade and compare learning algorithms. The first section describes how optimal power flow solutions are used to clear offers submitted to a simulated power exchange auction market. The second section defines how market participants are modelled as agents that use the reinforcement learning algorithms to adjust their bidding behaviour. It explains the modular structure of a multi-agent system that coordinates interactions between the auction model and participant agents.

4.1 Electricity Market Model

A power exchange auction market, based on SmartMarket by Zimmerman (2010, p.92), is used in this thesis as a trading environment for comparing reinforcement learning algorithms. In each trading period the auction accepts offers to sell blocks of power from participating agents[1]. A clearing process begins by withholding offers above the price cap, along with those specifying non-positive quantities. Valid offers for each generator are sorted into non-decreasing order with respect to price and converted into corresponding generator capacities and piecewise linear cost functions (See Section 4.1.1 below). The newly configured units form an optimal power flow problem, the solution to which provides generator set-points and nodal marginal prices that are used to determine the proportion of each offer block that is cleared and the associated clearing price. The cleared offers determine each agent's revenue and hence the profit used as a reward signal.

A nodal marginal pricing scheme is used in which the price of each offer is

---

[1]A double-sided auction, in which bids to buy blocks of power may be submitted by agents associated with dispatchable loads, has also been implemented, but this feature is not used.

cleared at the value of the Lagrangian multiplier on the power balance constraint for the bus at which the offer's generator is connected. An alternative discriminatory pricing scheme may be used in which offers are cleared at the price at which they were submitted (pay-as-bid). The advanced auction types from MATPOWER that scale nodal marginal prices are not used, but could be used in a detailed study of pricing schemes.

4.1.1 Optimal Power Flow

Bespoke implementations of both the DC and AC optimal power flow formulations from MATPOWER are used in the auction clearing process. The trade-offs between DC and AC formulations have been examined by Overbye, Cheng, and Sun (2004). DC models were found suitable for most nodal marginal price calculations and are considerably less computationally expensive to solve. The AC optimal power flow formulation is used to examine the exploitation of voltage constraints, that are not part of the DC formulation.

As in MATPOWER, generator active power, and optionally reactive power, output costs may be defined by convex n-segment piecewise linear cost functions

$$c^{(i)}(p) = m_i p + b_i \qquad (4.1)$$

where p is the generator set-point for $p_i \leq p \leq p_{i+1}$ with $i = 1, 2, \ldots n$, m_i is the variable cost for segment i in \$/MWh where $m_{i+1} \geq m_i$ and $p_{i+1} > p_i$, and b_i is the y-intercept in \$, also for segment i.

Since these cost functions are non-differentiable, the constrained cost variable approach from H. Wang, Murillo-Sanchez, Zimmerman, and Thomas (2007) is used to make the optimisation problem smooth. For each generator j a helper cost variable y_j is added to the vector of optimisation variables. Figure 4.1 (Zimmerman, 2010, Figure5-3) illustrates how the additional inequality constraints

$$y_j \geq m_{j,i}(p - p_i) + c_i, \quad i = 1 \ldots n \qquad (4.2)$$

ensure that y_j lies on or above $c^{(i)}(p)$ as the objective function minimises the sum of cost variables for all generators:

$$\min_{\theta, V_m, P_g, Q_g, y} \sum_{j=1}^{n_g} y_j \qquad (4.3)$$

The extended optimal power flow formulations from MATPOWER with user-

Figure 4.1: Piecewise linear active power cost function with constrained cost variable minimsation illustrated.

defined cost functions and generator P-Q capability curves are not used, but could be applied in further development of this work.

4.1.2 Unit De-commitment

The optimal power flow formulations constrain generator set-points between upper and lower power limits. The output of expensive generators can be reduced to the lower limit, but they can not be completely shutdown. The online status of generators could be added to the vector of optimisation variables, but being Boolean the problems would be mixed-integer non-linear programs which are typically very difficult to solve.

To compute a least cost commitment and dispatch the unit de-commitment algorithm from Zimmerman (2010, p.57) is used. The algorithm involves shutting down the most expensive units until the minimum generation capacity is less than the total load capacity and then solving repeated optimal power flow problems with candidate generating units, that are at their minimum active power limit, deactivated. The lowest cost solution is returned when no further improvement can be made and no candidate generators remain.

50

4.2 Multi-Agent System

Market participants are modelled using PyBrain software agents that use reinforcement learning algorithms to adjust their behaviour (Schaul et al., 2010). Their interaction with the market is coordinated in multi-agent simulations, the structure of which is derived from PyBrain's single player design.

This section describes: discrete and continuous market *environments*, agent *tasks* and *modules* used for policy function approximation and storing state-action values or action propensities. The process by which each agent's policy is updated by a *learner* is explained and the sequence of interactions between multiple agents and the market is described and diagrammed.

4.2.1 Market Environment

Each agent has a portfolio of n_g generators associated their environment. Figure 4.2 illustrates the association and how the environment references an instance of the auction market for offer submission. Each environment is responsible for (i) returning a vector representation of its current state and (ii) accepting an action vector which transforms the environment into a new state. To facilitate testing of value function based and policy gradient learning methods, both discrete and continuous representations of an electric power trading environment are defined.

Discrete Market Environment

An environment with n_s discrete states and n_a discrete action possibilities is defined for agents operating learning methods that make use of look-up tables. The environment produces a state s, where $s \in \mathbb{Z}^+$ and $0 \leq s < n_s$, at each simulation step and accepts an action a, where $a \in \mathbb{Z}^+$ and $0 \leq a < n_a$.

To keep the size of the state space reasonable, discrete states are derived only from the total system demand $d = \sum P_d$ where P_d is the vector of active power demand at each bus. Informally, the state space is n_s states between the minimum and maximum demand and the current state for the environment is the index of the state to which the current demand relates. Each simulation episode of n_t steps has a demand profile vector U of length n_t, where each element $0 \leq u_i \leq 1$. The load at each bus $P_{dt} = u_t P_{d0}$ in simulation period t, where P_{d0} is the initial demand vector. The state size $d_s = d(\max U - \min U)/n_s$ and the state space vector is $\mathcal{S} = d_s i$ for $i = 1 \ldots n_s$. At simulation step t, the state returned by the environment $s_t = i$ if $\mathcal{S}_i \leq P_{dt} \leq \mathcal{S}_{i+1}$ for $i = 0 \ldots n_s$.

```
┌─────────────────────────────┐              ┌─────────────────────────────┐
│     MarketEnvironment       │              │         SmartMarket         │
├─────────────────────────────┤              ├─────────────────────────────┤
│ name: string                │              │ auctionType: string         │
│ numOffbids: int = 1         │   market     │ priceCap: float = 500       │
│ offbidQty: bool = False     │   1..1       │ period: float               │
│ markups: tuple              │─────────────▶│ limits: dict                │
├─────────────────────────────┤              │ offers: list                │
│ getSensors()                │              │ bids: list                  │
│ performAction(action)       │              ├─────────────────────────────┤
│ reset()                     │              │ run()                       │
└─────────────────────────────┘              │ getOffbids(generator)       │
      env ▲          ╲                        └─────────────────────────────┘
      1..1 │           ╲  generators
            │            ╲ 1..*          ┌─────────────────────────────┐
                          ╲              │          Generator          │
┌─────────────────────────┐  ╲           ├─────────────────────────────┤
│        ProfitTask       │   ╲          │ name: string                │
├─────────────────────────┤    ▶         │ v_magnitude                 │
│ name: string            │              │ p: float = 100              │
│ sensor_limits: list     │              │ p_max: float = 200          │
│ actor_limits: list      │              │ p_min: float                │
├─────────────────────────┤              │ q: float                    │
│ getObservation()        │              │ q_max: float = 100          │
│ performAction(action)   │              │ q_min: float = -100         │
│ getReward()             │              │ p_cost: list                │
└─────────────────────────┘              │ q_cost: list                │
                                         ├─────────────────────────────┤
                                         │ is_load(): bool             │
                                         │ total_cost(p): float        │
                                         │ offers_to_pwl(offers)       │
                                         │ bids_to_pwl(bids)           │
                                         └─────────────────────────────┘
```

Figure 4.2: Agent environment UML class diagram.

Table 4.1: Example discrete action domain.

a	m_1	w_1	m_2	w_2
0	0	0	0	0
1	0	0	10	0
2	0	0	20	0
3	10	0	0	0
4	10	0	10	0
5	10	0	20	0
6	20	0	0	0
7	20	0	10	0
8	20	0	20	0

The action space for a discrete environment is defined by a vector m, where $0 \leq m_i \leq 100$, of percentage markups on marginal cost with length n_m, a vector w, where $0 \leq w_i \leq 100$, of percentage capacity withholds with length n_w and a scalar number of offers n_o, where $n_o \in \mathbb{Z}^+$, to be submitted for each generator associated with the environment.

A $n_a \times 2n_g n_o$ matrix with all permutations of markup and withhold for each offer that is to be submitted for each generator is computed. As an example, Table 4.1 shows all possible actions when markups are restricted to 0, 10% or 20%, $m = \{0, 10, 20, 30\}$, and 0% of capacity may be withheld, $w = \{0\}$, from two generators, $n_g = 2$, with one offer submitted for each, $n_o = 1$. Each row corresponds to an action and the column values specify the percentage price markup and the percentage of capacity to be withheld for each of the $n_g n_o$ offers. The size of the permutation matrix grows rapidly as n_o, n_g, n_m and n_w increase.

Continuous Market Environment

A continuous market environment that outputs a state vector s, where $s_i \in \mathbb{R}$, and accepts an action vector a, where $a_i \in \mathbb{R}$, is defined for agents operating policy gradient methods. Scalar variables m_u and w_u define the upper limit on the percentage markups on marginal cost and the upper limit on the percentage of capacity that can be withheld, respectively. Again, n_o defines the number of offers to be submitted for each generator associated with the environment.

The state vector can be any set of variables from the power system or market model. For example: bus voltages, branch power flows, generator limit Lagrangian multipliers etc. Each element of the vector provides one input to the neural network used for policy function approximation.

The action vector a has length $2n_g n_o$. Element a_i, where $0 \leq a_i \leq m_u$, corresponds to the percentage price markup and each element a_{i+1}, where $0 \leq a_{i+1} \leq w_u$, to the percentage of capacity to be withheld for the $(i/2)^{th}$ offer, where $i = 0, 2, 4, \ldots, 2n_g n_o$.

Not having to discretize the state space and compute a matrix of action permutations greatly simplifies the implementation of a continuous environment and increases in n_g and n_o only impact the number of output nodes on the neural network.

4.2.2 Agent Task

To allow alternative goals (such a profit maximisation or the meeting some target level for plant utilisation) to be associated with a single type of environment, an agent does not interact *directly* with its environment, but is paired with a particular *task*. A task defines the reward returned to the agent and thus defines the agent's purpose.

For all simulations in this thesis the goal of each agent is to maximise direct financial profit. Rewards are defined as the sum of earnings from the previous period t as determined by the difference between the revenue from cleared offers and the generator marginal cost at its total cleared quantity. Using some measure of risk adjusted return (as in (Moody & Saffell, 2001)) might be of interest in the context of simulated electricity trade and this would simply involve the definition of a new task and would not require any modification of the environment.

Agents with policy-gradient learning methods approximate their policy functions using artificial neural networks that are presented with an input vector s_n of length n_s where $s_{n,i} \in \mathbb{R}$. To condition the environment state before input to the connectionist system, where possible, a vector s_l of lower sensor limits and a vector s_u of upper sensor limits is defined. These are used to calculated a normalised state vector

$$v = 2 \left(\frac{s - s_l}{s_u - s_l} \right) - 1 \tag{4.4}$$

where $-1 \leq s_{n,i} \leq 1$.

The output from the policy function approximator y is denormalized using vectors of minimum and maximum action limits, a_{min} and a_{max} respectively, to give an action vector

$$a = \left(\frac{y + 1}{2} \right) (a_u - a_l) + a_l \tag{4.5}$$

where $0 \leq a_i \leq m_u$ and $0 \leq a_{i+1} \leq w_u$ for $i = 0, 2, 4, \ldots, 2n_g n_o$.

LearningAgent

learning: bool
logging: bool

getAction()
giveReward(reward)
integrateObservation(obs)
newEpisode()
learn()
reset()

ReinforcementDataSet

learning: bool
logging: bool

addSample(state, action, reward)
learn()

history
1..1

VariantRothErev

batchMode: bool
offPolicy: bool
experimentation: float
recency: float

learn()

learner
0..1

module
1..1

ActionValueTable

numActions: int
numStates: int

updateValue(row, column, value)
getValue(row, column)
getMaxAction(state)
initialize(value)

explorer
0..1

BoltzmannExplorer

epsilon: float = 2.0
decay: float = 0.9995

activate()

Figure 4.3: Learning agent UML class diagram.

4.2.3 Market Participant Agent

Each agent is defined as an entity capable of producing an action a based on a previous observation s of its environment. The UML class diagram in Figure 4.3 illustrates how each agent in PyBrain is associated with a *module*, a *learner* (modified Roth-Erev in the diagram), a *dataset* and an *explorer*.

The module is used to determine the agent's policy for action selection and returns an action vector a when activated with a state vector. When using value function based methods the module is a $n_s \times n_a$ table of the form

$$
\begin{array}{c}
 & \begin{array}{cccc} a_0 & a_1 & & a_{n_a} \end{array} \\
\begin{array}{c} s_0 \\ s_1 \\ \vdots \\ s_{n_s} \end{array} &
\left[\begin{array}{cccc}
v_{0,0} & v_{0,1} & \cdots & v_{0,m} \\
v_{1,0} & \ddots & & \vdots \\
\vdots & & \ddots & \vdots \\
v_{n,0} & \cdots & \cdots & v_{n_s,n_a}
\end{array} \right]
\end{array}
\tag{4.6}
$$

where each element $v_{i,j}$ is the value in state i associated with selecting action j. When using a policy gradient method, the module is a multi-layer feed-forward

55

artificial neural network that outputs a vector a when presented with observation s_n.

The learner can be any reinforcement learning algorithm that modifies the values/propensities/parameters of the module to increase expected future reward. The dataset stores state-action-reward triples for each interaction between the agent and its environment. The stored history is used by a learners when computing updates to the module.

Each learner has an association with an explorer that returns an explorative action a_e when activated with action a from the module. Softmax and ϵ-greedy explorers are implemented for discrete action spaces. Policy gradient methods use a module that adds Gaussian noise to a_m. The explorer has a parameter σ that relates to the standard deviation of the normal distribution. The actual standard deviation

$$\sigma_e = \begin{cases} \ln(\sigma + 1) + 1 & \text{if } \sigma \geq 0 \\ \exp(\sigma) & \text{if } \sigma < 0 \end{cases} \tag{4.7}$$

to prevent negative σ values from causing an error if automatically adjusted during back-propagation.

4.2.4 Simulation Event Sequence

Each simulation consists of one or more task-agent pairs. Figure 4.4 shows the class associations for a simulation experiment. At the beginning of each simulation step (trading period) t the market is initialised and all previous offers are removed. Figure 4.5 is a UML sequence diagram that illustrates the process of choosing and performing an action that follows. For each task-agent tuple an observation s_t is retrieved from the task and integrated into the agent. When an action is requested from the agent its module is activated with s_t and the action $a_{e,t}$ is returned. Action $a_{e,t}$ is performed on the environment associated with the agent's task.

When all actions have been performed the offers are cleared by the market using the solution to a newly formed optimal power flow problem. Figure 4.6 illustrates the subsequent reward process. The cleared offers associated with the generators in the task's environment are retrieved from the market and the reward r_t is computed from the difference between revenue and marginal cost at the total cleared quantity. The reward r_t is given to the associated agent and the value is stored, along with the previous state s_t and selected action $a_{e,t}$, under a new sample is the dataset.

Figure 4.4: Market experiment UML class diagram.

The learning process is illustrated by the UML sequence diagram in Figure 4.7. Each agent learns from its actions using r_t, at which point the values or parameters of the module associated with the agent are updated according to the output of the learner's algorithm. Each agent is then reset and the history of states, actions and rewards is cleared.

The combination of an action, reward and learning process for each agent constitutes one step of the simulation and the processes are repeated until a specified number of steps are complete.

4.3   Summary

The power exchange auction market model defined in this chapter provides a layer of abstraction over the underlying optimal power flow problem and presents agents with a simple interface for selling power. The modular nature of the simulation framework described allows the type of learning algorithm, policy function approximator, exploration technique or task to be easily changed. The framework can simulate competitive electric power trade using almost any conventional bus-branch power system model with little configuration, but provides the facilities for adjusting all of the main aspects of a simulation. The framework's modularity

Figure 4.5: Action selection sequence diagram.

Figure 4.6: Sequence diagram for the reward process.

Figure 4.7: Learning sequence diagram.

and support for easy configuration is intended to allow transparent comparison of learning methods under a wide variety of different scenarios.

Chapter 5

Nash Equilibrium Analysis

This chapter presents a simulation that examines a system of agents competing to sell electricity and its convergence to a Nash equilibrium. Value function based and policy gradient reinforcement learning algorithms are compared in their convergence to an optimal policy using a six bus electric power system model.

5.1 Introduction

This thesis presents the first case of policy gradient reinforcement learning methods being applied to simulated electricity trade. As a first step it is necessary to confirm that when using these methods, a system of multiple agents will converge to the same Nash equilibrium[1] that a traditional closed-form simulation would produce.

This is the same approach used by Krause et al. (2006) before performing the study of congestion management techniques that is reviewed in Section 3.2.2. Nash equilibria can be difficult to determine in complex systems so the experiment presented here utilises a model simple enough that it can be determined through exhaustive search.

By observing actions taken and rewards received by each agent over the initial simulation periods it is possible to compare the speed and consistency with which different algorithms converge to an optimal policy. In the following sections the objectives of the simulation are defined, the simulation setup is explained and plots of results, with discussion and critical analysis, are provided.

---

[1]Informally, a Nash equilibrium is a point in a non-cooperative game at which no player is motivated to deviate from its strategy, as it would result in lower gain (Nash, 1950, 1951).

5.2 Aims and Objectives

Some elements of the simulations reported in this chapter are similar to those presented by Krause et al. (2006). One initial aim of this work is to reproduce their findings as a means of validating the approach. The additional objectives are to show:

- That policy gradient methods converge to the same Nash equilibrium as value function based methods and traditional closed-form simulations,

- Some the characteristics of policy gradient methods and how they differ from value function based methods.

Meeting these objectives aims to provide a basis for using policy gradient methods in more complex simulations, to show that they can *learn to trade power* and to provide guidance for algorithm parameter selection.

5.3 Method of Simulation

Learning methods are compared in this chapter by repeating the same simulation and switching the algorithms used by the agents. An alternative might be to use a combination of methods in the same simulation, but the approach used here is intended to be an extension of the work by Krause et al. (2006).

Each simulation uses a six bus electric power system model adapted from Wood and Wollenberg (1996, pp. 104, 112, 119, 123-124, 549). The model provides a simple environment for electricity trade with a small number of generators and branch flow constraints that slightly increase the complexity of the Nash equilibria. The buses are connected by eleven transmission lines at 230kV. The model contains three generating units with a total capacity of 440MW and loads at three locations, each 70MW in size. The connectivity of the branches and the locations of the generators and loads is shown in Figure 5.1. Data for the power system model was taken from a case provided with MATPOWER and is listed in Appendix B.1.

Two sets of quadratic generator operating cost functions, of the form $c(p_i) = ap_i^2 + bp_i + c$ where p_i is the output of generator i, are defined in order to create two different equilibria for investigation. The coefficients a, b and c for cost configuration 1 are listed in Table 5.1. This configuration defines two low cost generators that can not offer a price greater than the marginal cost of the most expensive generator when they apply the maximum possible markup. The set

Figure 5.1: Single-line diagram for six bus power system model.

Table 5.1: Generator cost configuration 1.

Gen	a	b	c
1	0.0	4.0	200.0
2	0.0	3.0	200.0
3	0.0	6.0	200.0

of coefficients for cost configuration 2 is listed in Table 5.2. This configuration narrows the cost differences such that offer prices may overlap and may exceed the marginal cost of the most expensive generator.

As in Krause et al. (2006), no load profile is defined. The system load is assumed to be peak in all periods and only one state is defined for methods using look-up tables. Each simulation step is assumed to be one hour in length.

For all generators $P^{min} = 0$ so as to simplify the equilbria and avoid the need for the unit de-commitment algorithm. The maximum capacity for the most expensive generator $P_3^{max} = 220$MW such that it may supply all of the load if dispatched, subject to branch flow limits. This generator is associated with a passive agent that always offers full capacity at marginal cost. For the less expensive generators $P_1^{max} = P_2^{max} = 110$MW. These two generators are each associated with an active learning agent whose activity in the market is restricted

Table 5.2: Generator cost configuration 2.

Gen	a	b	c
1	0.0	5.1	200.0
2	0.0	4.5	200.0
3	0.0	6.0	200.0

to one offer of maximum capacity in each period, at a price representing a markup of between 0 and 30% on marginal cost. Methods restricted to discrete actions may markup in steps of 10%, giving possible markup actions of 0, 10%, 20% and 30%. No capacity withholding is implemented. Discriminatory pricing (pay-as-bid) is used in order to provide a clearer reward signal to agents with low cost generators.

The algorithms compared are: Q-learning, ENAC, REINFORCE and the modified Roth-Erev technique (See Section 2.4). Default algorithm parameter values from PyBrain are used and no attempt is made to study parameter sensitivity or variations in function approximator design.

For the Q-learning algorithm $\alpha = 0.3$, $\gamma = 0.99$ and ϵ-greedy action selection is used with $\epsilon = 0.9$ and $d = 0.98$. For the Roth-Erev technique $\epsilon = 0.55$, $\phi = 0.3$ and Boltzmann action selection is used with $\tau = 100$ and $d = 0.99$.

Both REINFORCE and ENAC use a two-layer neural network with one linear input node, one linear output node, no bias nodes and with the connection weight initialised to zero. This is a typically PyBrain configuration taken from the supplied examples. A two-step episode is defined for the policy gradient methods and five episodes are performed per learning step. The exploration parameter σ for these methods is initialised to zero and adjusted manually after each episode such that:

$$\sigma_t = d(\sigma_{t-1} - \sigma_n) + \sigma_n \qquad (5.1)$$

where $d = 0.998$ is a decay parameter and $\sigma_n = -0.5$ specifies the value that is converged to asymptotically. In each simulation the learning rate $\gamma = 0.01$ for the policy gradient methods, apart from for ENAC under cost configuration 2 where $\gamma = 0.005$. Both active agents use the same parameter values in each simulation.

As in Krause et al. (2006), the point of Nash equilibrium is established by computing each agent's reward for all possible combinations of discrete markup. The rewards for Agent 1 and Agent 2 under cost configuration 1 are given in Table 5.3. The Nash equilibrium points are marked with a *. The table shows that the optimal policy for each agent is to apply the maximum markup to each offer as

Table 5.3: Agent rewards under cost configuration 1

		G_1							
		0.0%		10.0%		20.0%		30.0%	
		r_1	r_2	r_1	r_2	r_1	r_2	r_1	r_2
	0.0%	0.0	0.0	40.0	0.0	80.0	0.0	120.0	0.0
G_2	10.0%	0.0	33.0	40.0	33.0	80.0	33.0	120.0	33.0
	20.0%	0.0	66.0	40.0	66.0	80.0	66.0	120.0	66.0
	30.0%	0.0	99.0	40.0	99.0	80.0	99.0	120.0*	99.0*

Table 5.4: Agent rewards under cost configuration 2

		G_1							
		0.0%		10.0%		20.0%		30.0%	
		r_1	r_2	r_1	r_2	r_1	r_2	r_1	r_2
	0.0%	0.0	0.0	51.0	0.0	0.0	0.0	0.0	0.0
G_2	10.0%	0.0	49.5	51.0	49.5	0.0	49.5	0.0	49.5
	20.0%	0.0	92.2	51.0	99.0	0.0	99.0	0.0	99.0
	30.0%	0.0	126.8	54.8*	138.4*	0.0	148.5	0.0	148.5

their generators are always dispatched. The rewards under cost configuration 2 are given in Table 5.4. This table shows that the optimal point occurs when Agent 2 applies its maximum markup and Agent 1 offers a price just below the marginal cost of the passive agent's generator.

5.4 Simulation Results

Each action taken by an agent and the consequent reward is recorded for each simulation. Values are averaged over the ten simulation runs and standard deviations are calculated using the formula

$$SD = \sqrt{\frac{1}{N-1} \sum_{i=0}^{N} (x_i - \bar{x})^2} \qquad (5.2)$$

where x_i is the action or reward value in simulation i of N simulation runs and $\bar{x}$ is the mean of the values.

Figure 5.2 shows the average markup on marginal cost and the standard deviation over the ten simulation runs for Agent 1 under price configuration 1, using the four learning methods. The second y-axis in each plot relates to the exploration parameter for each method. Figure 5.3 shows the same information

for Agent 2. Plots of reward are not given as generator prices and the market are configured such that an agent's reward is directly proportional to its action. The plots are vertically aligned and have equal x-axis limits to assist algorithm comparison.

Figures 5.4 and 5.5 plot the average markup and reward over ten simulation runs for Agent 1 and Agent 2, respectively, under price configuration 2 for the variant Roth-Erev, Q-learning learning methods. The plots for REINFORCE and ENAC in these figures are for actual values in one simulation run as the number of interactions and variation in values makes the results difficult to observe otherwise.

5.5 Discussion and Critical Analysis

Under cost configuration 1 the agents face a relatively simple control task and receive a clear reward signal that is directly proportional to their markup. The results show that all of the methods consistently converge to the point of Nash equilibrium. The variant Roth-Erev method shows very little variation around the mean once converged, due to the use of Boltzmann exploration with a then low temperature parameter value. The constant variation around the mean that can be seen for Q-learning once converged is due to the use of ϵ-greedy action selection and can be removed if a Boltzmann explorer is selected.

Empirical studies have also shown that the speed of convergence is largely determined by the rate at which the exploration parameter value is reduced. However, the episodic nature of the policy gradient methods requires them to make several interactions per learning step and therefore a larger number of initial exploration steps are required. Policy gradient methods have also been found to be highly sensitive to the choice of learning rate. High values cause large changes to policy parameters to be made at each step and may cause the algorithm to not converge, but low values cause the algorithm to learn very slowly.

Cost configuration 2 provides a more challenging control problem in which Agent 1 must learn to undercut the passive agent. The results show that the variant Roth-Erev and Q-learning methods both consistently learn their optimal policy and converge to the Nash equilibrium. However, there is space for Agent 1 to markup its offer by slightly more than 10% and still undercut the passive agent, but the methods with discrete actions are not able to exploit this and do not receive the small additional profit.

The results for the policy gradient methods under cost configuration 2 show
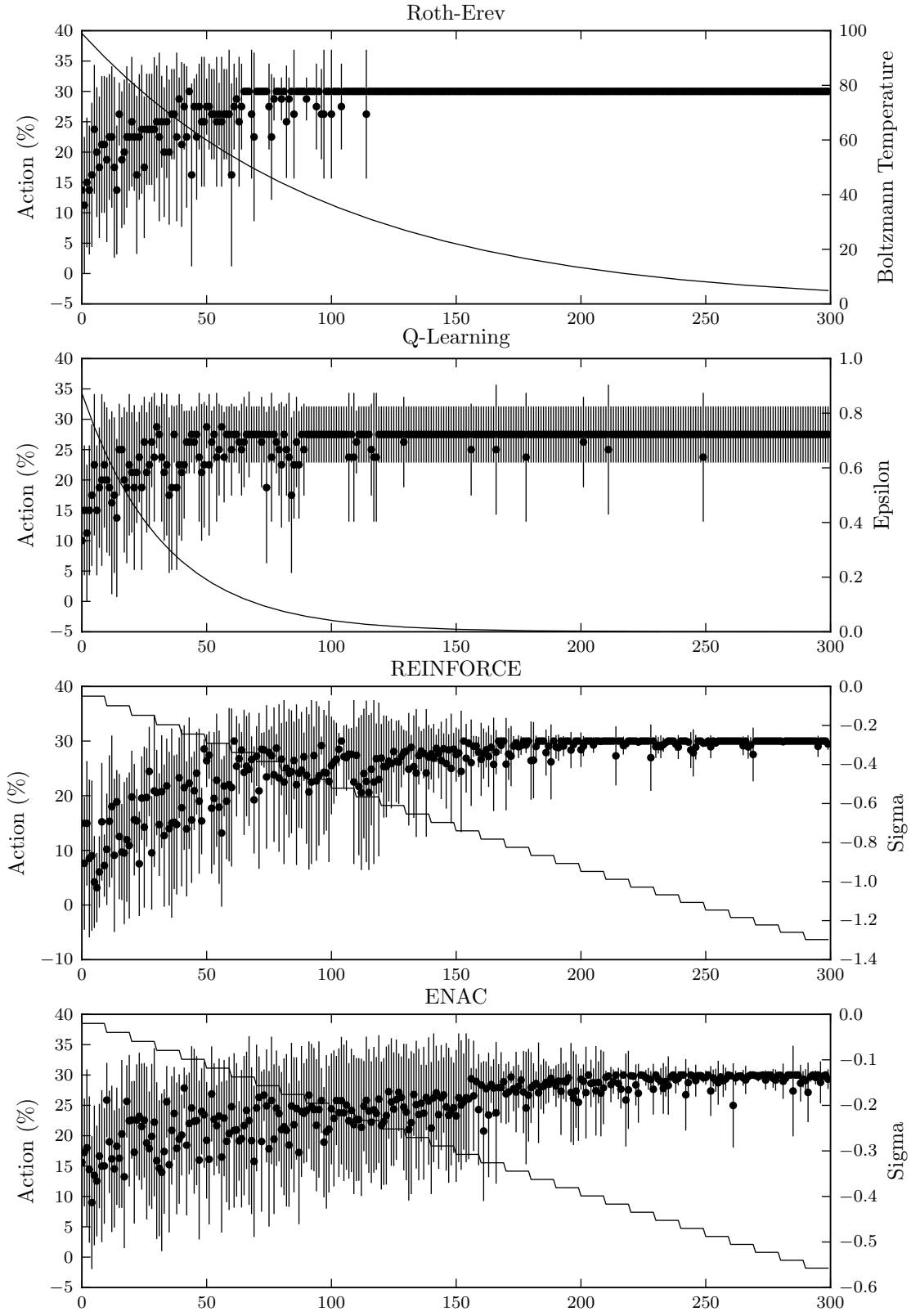
Figure 5.2: Average markup and standard deviation for Agent 1 under cost configuration 1.
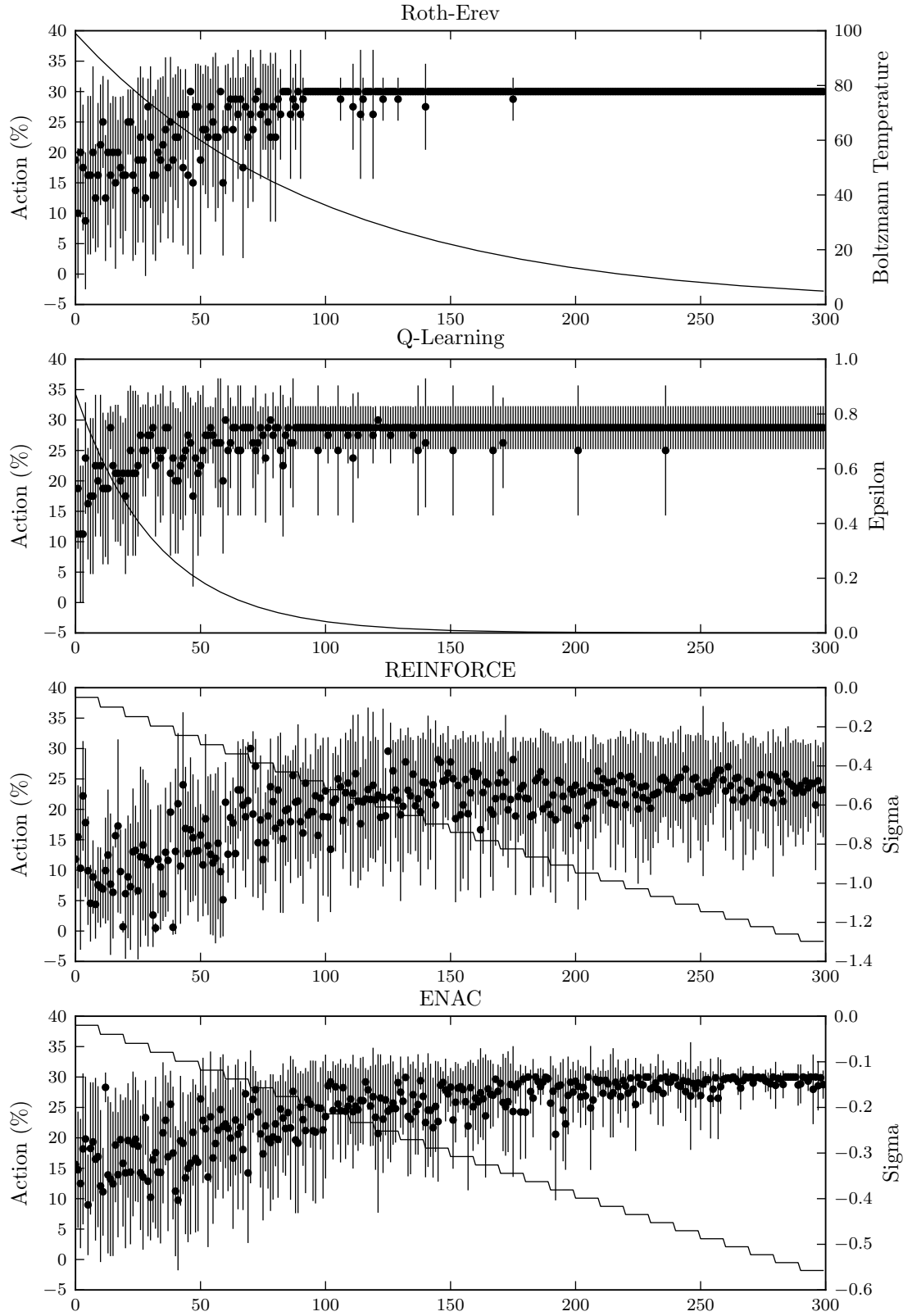
Figure 5.3: Average markup and standard deviation for Agent 2 under cost configuration 1.
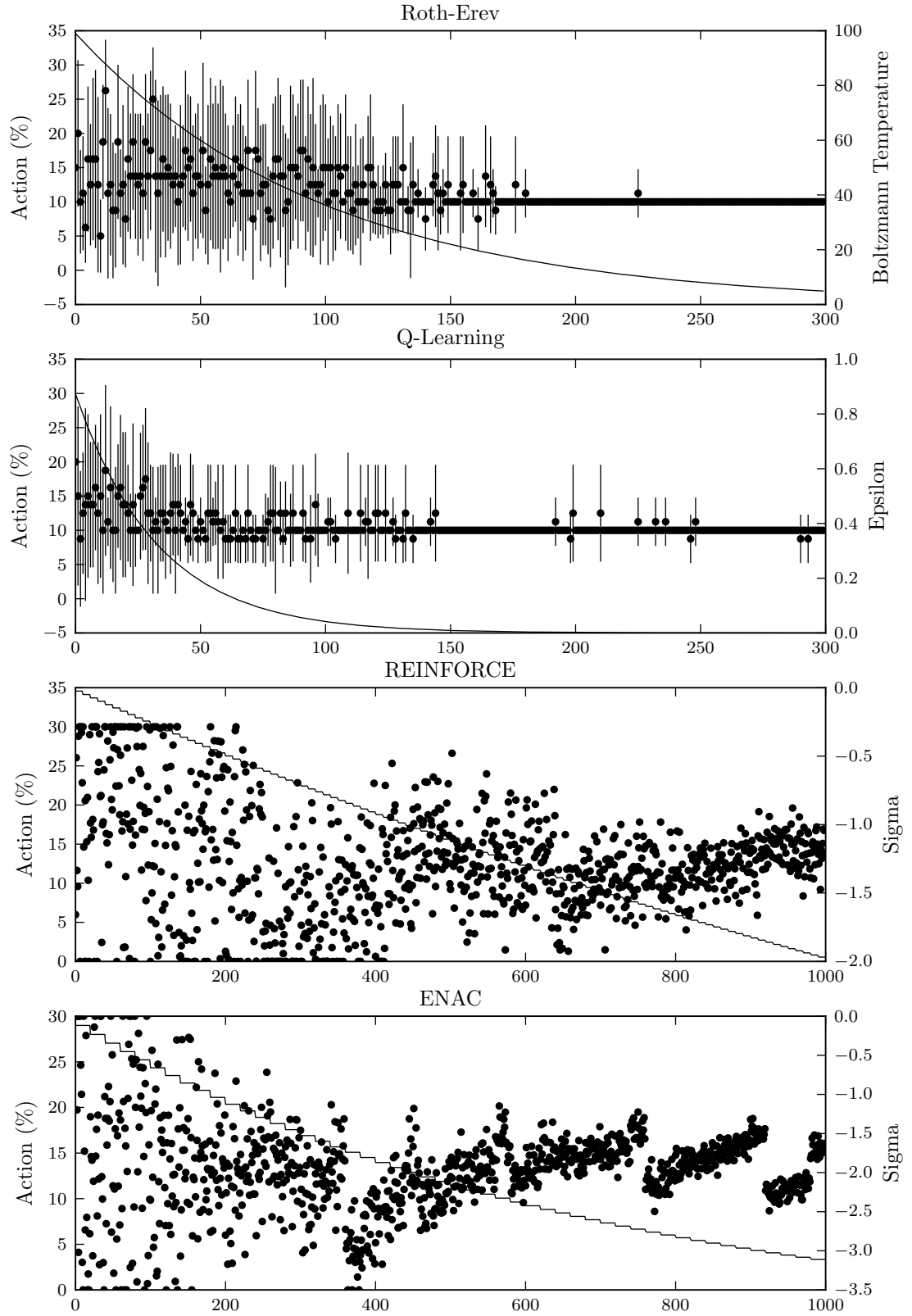
Figure 5.4: Average markup and standard deviation for Agent 1 under cost configuration 2.
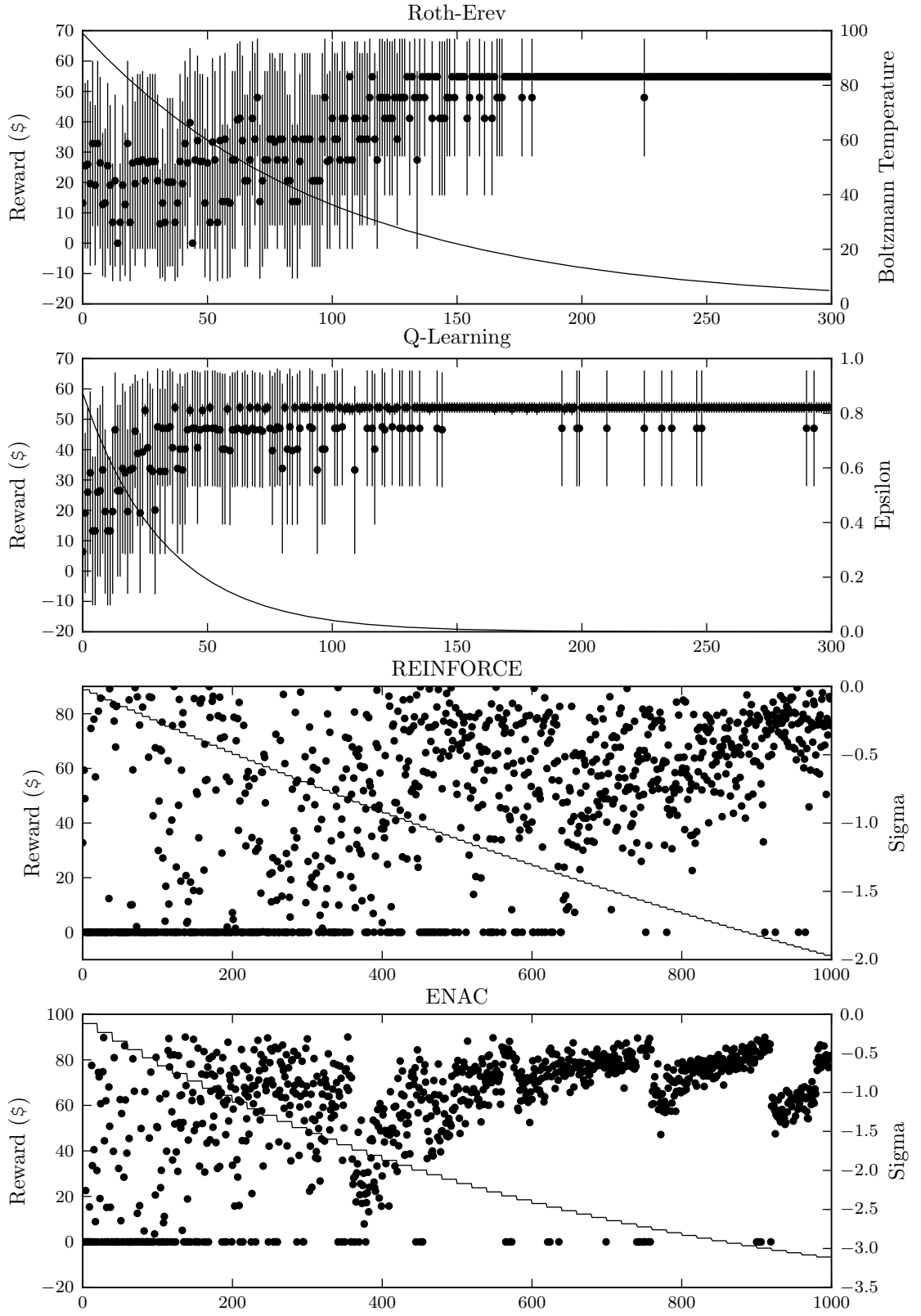
Figure 5.5: Average reward and standard deviation for Agent 1 under cost configuration 2.

that they learn to reduce their markup if their offer price starts to exceed that of the passive agent and the reward signal drops. However, a chattering effect below the Nash equilibrium point can be clearly seen for ENAC and the method does not learn to always undercut the other agent. These methods require a much larger number of simulation steps and for the exploration parameter to decay slowly if they are to produce this behaviour. This is due to the need for a lower learning rate that ensures fine policy adjustments can be made and for several interactions to be performed between each learning step.

5.6   Summary

By observing the state to which a multi-learning-agent system converges, it is possible to verify that learning algorithms produce the same Nash equilibrium that closed-form simulations provide. The results presented in this chapter closely correspond with those from Krause et al. (2006) for Q-learning and show equivalent behaviour for the variant Roth-Erev method. The simulations illustrate how challenging unsupervised learning is in a continuous environment, even for a simple problem. Tasks in which a large reward change can occur for a very small change in policy prove difficult for policy gradient methods to learn and require low learning rates and lengthy periods of exploration. The operation of policy gradient methods with noisy, multi-dimensional state data is not examined in this chapter and deserves investigation.

Chapter 6

System Constraint Exploitation

This chapter explores the exploitation of constraints by learning agents in a dynamic electricity trading environment. Value function based and policy gradient reinforcement learning methods are compared using a modified version of the IEEE Reliability Test System.

6.1   Introduction

Having examined the basic learning characteristics of four algorithms in Chapter 5, this chapter extends the approach to examine their operation in a complex dynamic environment. It explores the ability of policy gradient methods to operate with multi-dimensional, continuous state and action data in the context of *learning to trade power*.

A reference electric power system model from the IEEE Reliability Test System (RTS) (Application of Probability Methods Subcommittee, 1979) provides a realistic environment for agents to compete with diverse portfolios of generating plant supply dynamic loads. System constraints change as agents adjust their strategies and loads follow a hourly profile that is varied in shape from day-to-day over the course of a simulated year. By observing average profits at each hour of the day, the ability of methods to successfully observe and exploit constraints is examined.

6.2   Aims and Objectives

This experiment aims to compare policy gradient and traditional reinforcement learning methods in a dynamic electricity trading environment. Specifically, the objectives are to determine:

- If the policy gradient methods can achieve greater profitability under dynamic system constraints using a detailed state vector.

- The value of using an AC optimal power flow formulation in agent based electricity market simulation.

Meeting the first objective aims to demonstrate some of the value of using policy gradient methods in electricity market participant modelling and to determine if they warrant further research in this domain.

6.3 Method of Simulation

Learning methods are again compared by repeating simulations of competitive electricity trade switching the algorithms used by the competing agents. Some simplification of the state and action representations for value function based methods is required, but generation portfolios and load profiles are identical for each algorithm test.

The RTS has 24 bus locations that are connected by 32 transmission lines, 4 transformers and 2 underground cables. The transformers tie a 230kV area to a 138kV area. The original model has 32 generators of 9 different types with a total capacity of 3.45GW. To reduce the size of the discrete action space, five 12MW and four 20MW generators are removed. This is deemed reasonable as it reduced the number of generators by 28%, but their combined capacity is only 4.1% of the original total generation capacity and the remainder is more than sufficient to meet demand. To further reduce action space sizes all generators of the same type at the same bus are aggregated into one generating unit. This can be considered to be the representation of each individual power station in the market, rather than each alternator stage. The model has loads at 17 locations and the total demand at system peak is 2.85GW.

Again, generator marginal costs are quadratic functions of output and are defined by the parameters in Table 6.1. Figure 6.1 shows the cost functions for each of the seven types of generator and illustrates their categorisation by fuel type. Generator cost function coefficients were taken from an RTS model by Georgia Tech Power Systems Control and Automation Laboratory[1] which assumes Coal costs of 1.5 \$/MBtu[2], Oil costs of 5.5 \$/MBtu and Uranium costs of 0.46 \$/MBtu. Data for the modified model is provided in Appendix B.2 and

---

[1]http://pscal.ece.gatech.edu/testsys/
[2]1 Btu $\approx$ 1055 Joules

Table 6.1: Generator types and cost parameters for the simplified IEEE Reliability Test System.

Code	a	b	c	Type
U50	0.0	0.0010	0.001	Hydro
U76	0.01414	16.0811	212.308	Coal
U100	0.05267	43.6615	781.521	Oil
U155	0.00834	12.3883	382.239	Coal
U197	0.00717	48.5804	832.758	Oil
U350	0.00490	11.8495	665.109	Coal
U400	0.00021	4.4231	395.375	Nuclear

Table 6.2: Agent portfolios.

Agent	U50 Hydro	U76 Coal	U100 Oil	U155 Coal	U197 Oil	U350 Coal	U400 Nuclear	Total (MW)
1		2×		1×			1×	707
2		2×		1×			1×	707
3	6×				3×			891
4			3×	2×		1×		960

the connectivity of branches and the location of generators and loads is illustrated in Figure 6.3.

The generating stock is divided into 4 portfolios (See Table 6.2) that are each endowed to a learning agent. Portfolios were chosen such that each agent has: a mix of base load and peaking plant, approximately the same total generation capacity and generators in different areas of the network. The generator labels in Figure 6.3 specify the associated agent. The synchronous condenser is associated with a passive agent that always offers 0 MW at 0 $/MWh (the unit can be dispatched to provide or absorb reactive power).

Markups on marginal cost are restricted to a maximum of 30% and discrete markups of 0, 15% or 30% are defined for value function based methods. Up to 20% of the total capacity of each generator can be withheld and discrete withholds of 0 or 20% are defined. Initially only one offer per generator is required, but this is increased to two in order to explore the effect of increased offer flexibility.

The environment state for all algorithm tests consists of a forecast of the total system demand for the next period. The system demand follows an hourly profile that is adjusted according to the day of the week and the time of year. The profiles are provided by the RTS and are illustrated in Figure 6.2. For tests
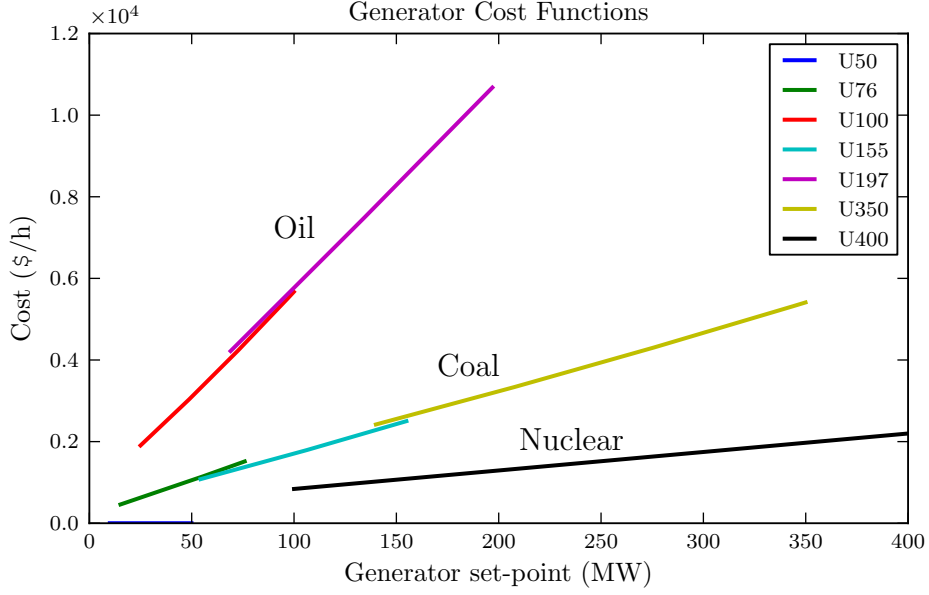
Figure 6.1: Generator cost functions for the IEEE Reliability Test System

of value function based methods and the Stateful Roth-Erev learning algorithm, the continuous state is divided into 3 discrete states of equal size, that allow differentiation between low, medium and peak demand.

To investigate the exploration of constraints, AC optimal power flow is used and the state vector for agents using policy gradient methods is optionally adjusted to combine the demand forecast with voltage constraint Lagrangian multipliers for all generator buses and the voltage magnitude at all other buses. Lagrangian multipliers are used for generator buses as generators typically fix the voltage at their associated bus. Branch flows are not included in the state vector as flow limits in the RTS are high and are typically not reached at peak demand. Generator capacity limits are binding in most states of the RTS, but the output of other generators is deemed to be hidden from agents.

The nodal marginal pricing scheme is used and cleared offer prices are determined by the Lagrangian multiplier on the power balance constraint for the bus at which the generator associated with the offer is connected.

Typical parameter values are used for each of the algorithms. Again, no attempt to study parameter sensitivity is made. Learning rates are set low and exploration parameters decay slowly due to the length and complexity of the simulation. For Q-learning $\alpha = 0.2$, $\gamma = 0.99$ and ϵ-greedy action selection is used with $\epsilon = 0.9$ and $d = 0.999$. For Roth-Erev learning $\epsilon = 0.55$, $\phi = 0.3$ and Boltzmann action selection is used with $\tau = 100$ and $d = 0.999$.
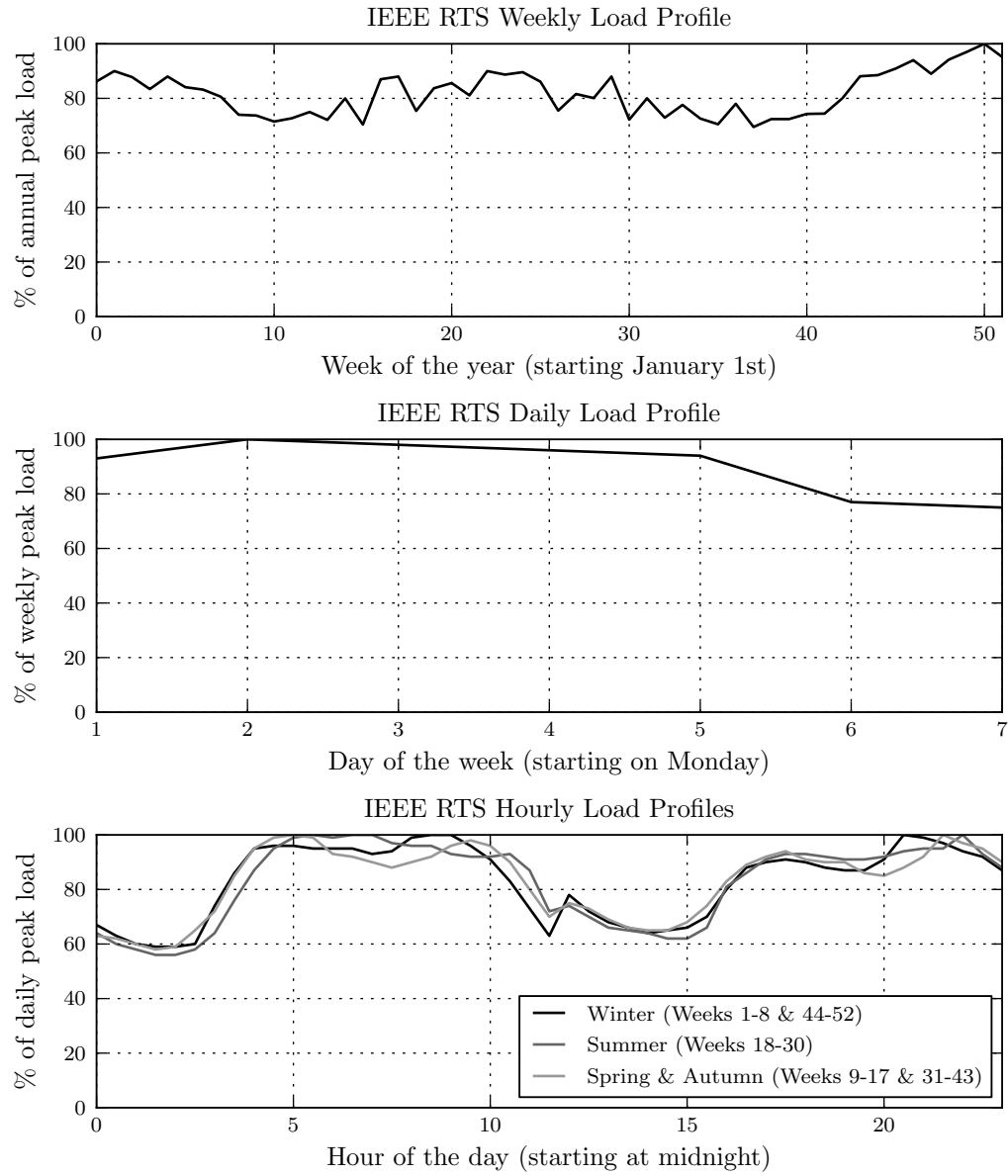
Figure 6.2: Hourly, daily and weekly load profile plots from the IEEE Reliability Test System

400 MW A1 ↑ 333 MW

Nuclear

Nuclear   Bus 18   A2 400 MW

Hydro, 6×U50

A3 300 MW

Bus 17

Bus 21   Bus 22

2×U155
1×U350

Bus 23   A4

A2 155 MW   660 MW

Bus 16

Bus 19   Bus 20

100
MW

181 MW   128 MW

591 MW

Bus 14
Bus 15   A5 Synch.
Cond.   3×U197   A3

A1   317 MW

155
MW   194 MW   230 kV   Bus 13

265 MW

Bus 24   Bus 11   Bus 12

Bus 3   Bus 9   Bus 10   Bus 6

195 MW

cable

108
MW

Bus 4   175 MW   136 MW

Bus 8

74 MW

Bus 5

171 MW ↓

↑ 125 MW

71 MW

Bus 7

138 kV

cable

2×U76 Bus 1   Bus 2   2×U76   A4 300 MW

152 MW A1   A2 152 MW

108
MW   97
MW   3×U100

Figure 6.3: IEEE Reliability Test System

77

Figure 6.4: Average rewards for Agent 1 and Agent 4, comparing the modified and Stateful Roth-Erev methods.

Again a typical PyBrain two-layer neural network configuration with linear input and output nodes, no bias nodes and randomised initial connection weights are used for policy function approximation. The initial exploration rate $\sigma = 0$ for both policy gradient methods and decays according to Equation (5.1) with $d = 0.995$ and $\sigma_n = -0.5$. Constant learning rates are used in each simulation with $\gamma = 0.01$ for REINFORCE and $\gamma = 0.005$ for ENAC.

6.4   Simulation Results

Each agent's rewards are recorded for a simulated year of 364 trading episodes, each consisting of 24 interactions. To compare algorithms, the average reward for each hour of the day is calculated for each agent and plotted. Only results for Agent 1 and Agent 4 are given as Agents 1 and Agent 2 have identical portfolios and most of Agent 3's portfolio consists of hydro-electric plant with zero cost. The method of applying percentage markups on marginal cost has not effect for generators with zero cost and almost identical results are found for all algorithms.
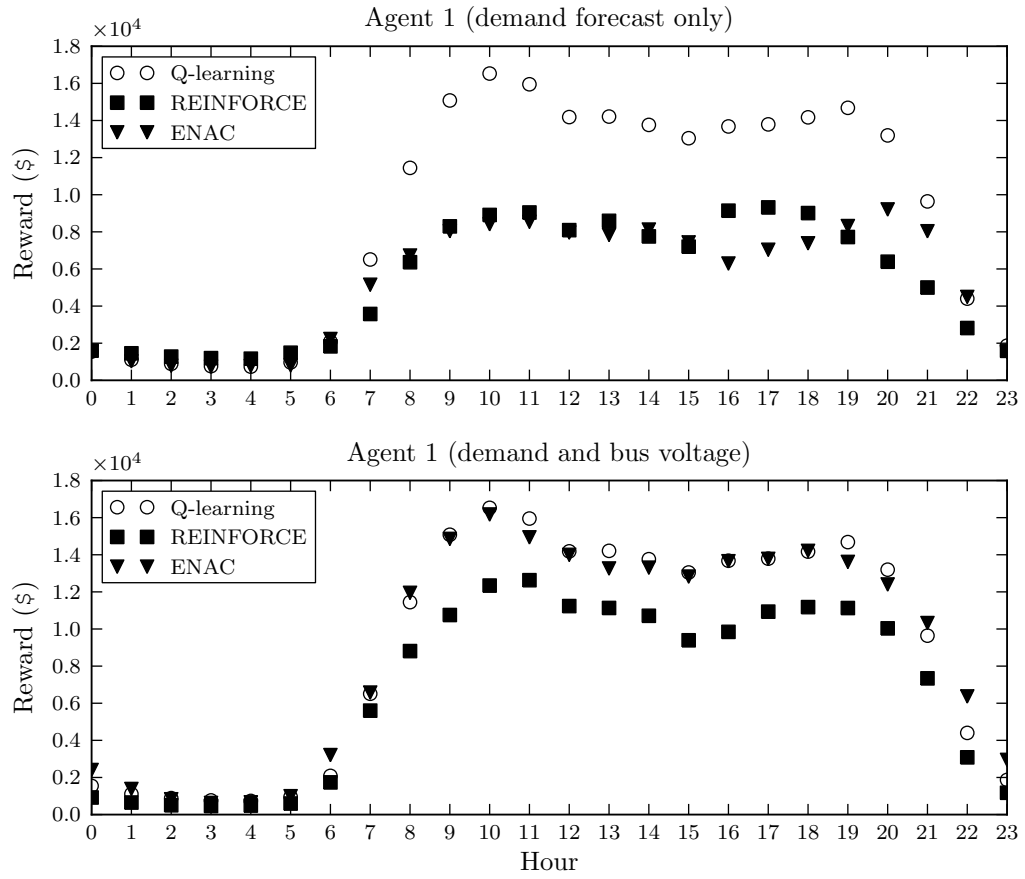
Figure 6.5: Average rewards for Agent 1 under two state configurations.

Figure 6.4 compares the modified Roth-Erev method with the Stateful Roth-Erev method. The plots show average rewards for Agent 1 and Agent 4 when using Q-learning and the two Roth-Erev variants.

Figure 6.5 and Figure 6.6 compare policy gradient methods under two state vector configurations. Figure 6.5 concerns Agent 1 and shows the average reward received for a state vector consisting solely of a demand forecast and for a combined demand forecast and bus voltage profile state vector. Figure 6.6 shows average rewards for Agent 4 under the same configurations.

Figure 6.7 shows average rewards for agents 1 and 4 from a repeat of the bus voltage profile state simulation, but with two offers required per generator. Due to time constraints and limited simulation resources only results for Q-learning and ENAC are given.
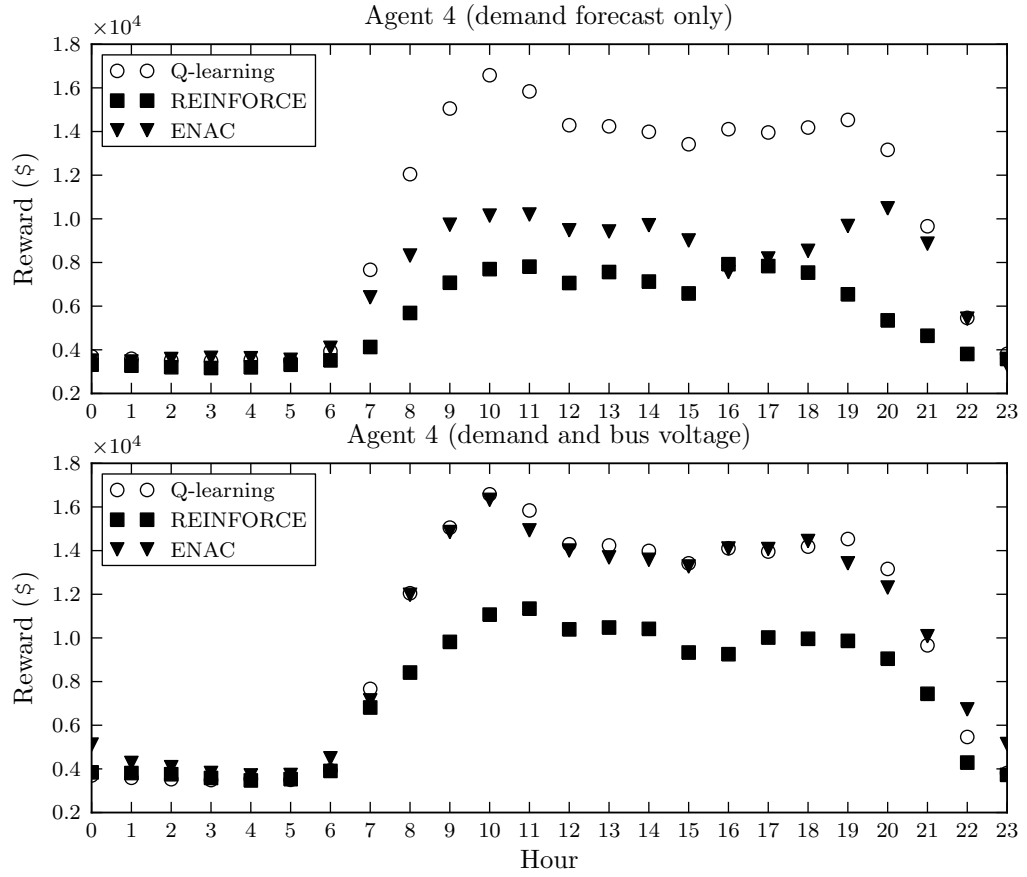
Figure 6.6: Average rewards for Agent 4 under two state configurations.
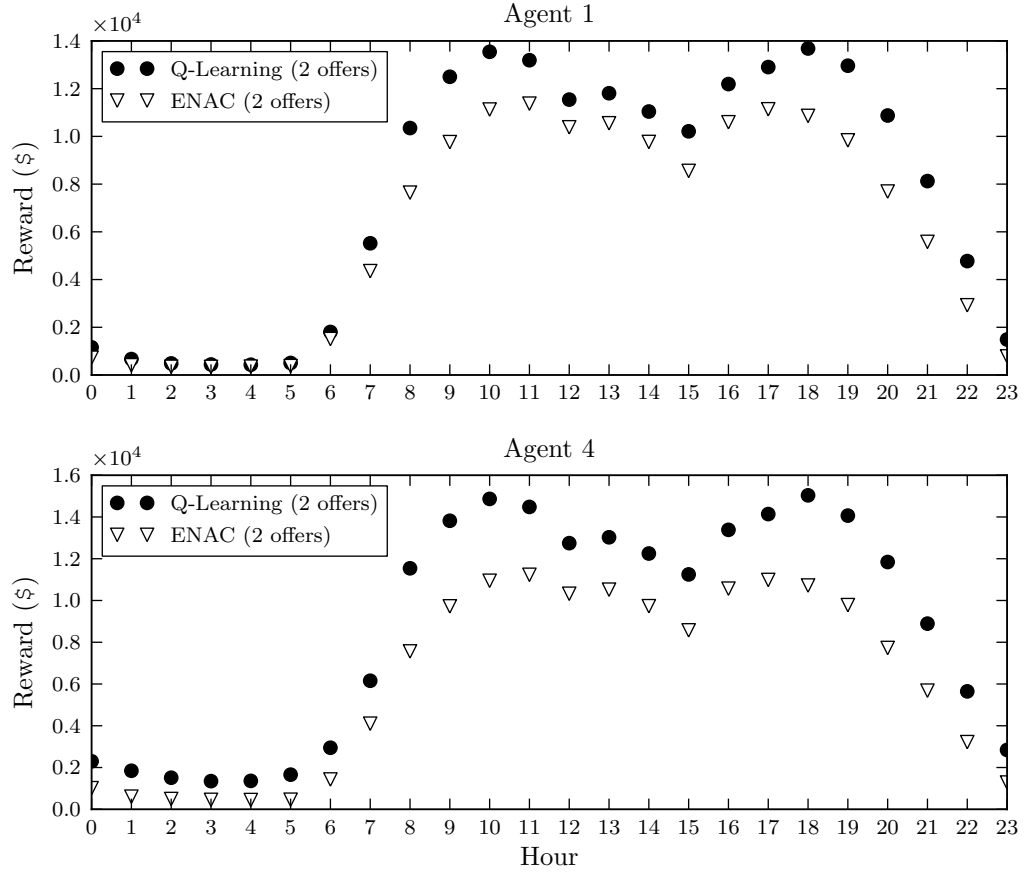
Figure 6.7: Average rewards for Agent 1 and Agent 4 with two offers submitted per generator.

6.5 Discussion and Critical Analysis

Agents with a discrete environment have 216 possible actions to choose from in each state when required to submit one offer per generator. Figure 6.4 shows that, using Q-learning, agents are able to learn an effective policy that yields increased profits under two different portfolios. The importance of utilising environment state data in a dynamic electricity setting is illustrated by the differences in average reward received by the modified Roth-Erev method and the Stateful Roth-Erev method. The optimal action for an agent depends upon the current system load and the stateless Roth-Erev formulation is unable to interpret this. The Stateful Roth-Erev method can be seen to achieve approximately the same performance as Q-learning.

Including bus voltage constraint data in the state for a discrete environment would result in a state space of impractical size, but including it in a continuous environment was straight-forward. The results show that ENAC achieves greater profits when presented with a combined demand forecast and bus voltage state vector. REINFORCE performs less well than ENAC, but also shows improvement over the pure demand forecast case. ENAC achieves equivalent, but not greater performance than Q-learning in all periods of the trading day when using the voltage data. ENAC is not able to use the additional state information to any further advantage, but does learn a profitable policy.

Simply changing the number of offers that are required to be submitted for each generator from 1 to 2, increases the number of discrete action possibilities in each state to 46,656. Figure 6.7 shows that Q-learning is still able to achieve a similar level of reward as in the one offer case. The profitability for both methods is degraded, but ENAC receives significantly lower average reward when the agent is required to produce an action vector of twice the size and is not able to use the increased flexibility in its offer structure to any advantage.

With state and action spaces on this scale, computing updates to an agent's look-up table or neural network adds considerably to the computational expense of the simulation. Researchers wishing to apply these methods in larger problems must be willing to investigate program optimisation and parallel or distributed processing. Studies not requiring this level of complexity are seemingly best using a state-value function based method, such as Q-learning or the Stateful Roth-Erev formulation.

The lack of involvement in the market from the hydro-electric power plant largely negates the participation of Agent 3 and exposes one significant shortcoming of the approach. This could be overcome by allowing markups in dollars,

rather than as a percentage of marginal cost.

Generation portfolios were configured such that agents would receive a mix of low-cost base-load plant and expensive peak-supply plant. However, the cost differences between fuel types are such that an offer of power from a coal or nuclear power station can not exceed in price that from a unit with a more expensive fuel type. Greater competition and more complex equilibria could introduced to the simulation if fuel cost differences were not as large or maximum markups on price were greater. In Rastegar et al. (2009), for example, a 300% markup limit is set.

The dynamics of this simulation could also be greatly increased by introducing demand-side participation. It could allow agents to directly influence the state of the environment, through the demand forecast sensor. It would also give agents more options for competition, increasing the complexity of their optimal policy and posing a greater challenge to the learning algorithms.

6.6   Summary

In this chapter policy gradient reinforcement learning algorithms have been applied in a complex dynamic electricity trading simulation and assessed in their ability to exploit constraints in the system. They were found to be a valid technique for *learning to trade power*, but were outperformed by Q-learning in most configurations of environment state and action space. This includes a simulation with action spaces that were expected to be too large for Q-learning to explore, but to be of no significant challenge to policy gradient methods.

The addition of bus voltage sensor data in the state vector of agents operating policy gradient methods was found to improve their performance. However, no evidence was found to suggest that they could use this information to increase their profitability above that of agents operating the Q-learning or Stateful Roth-Erev method. Indeed, it is believed that this can be considered a general finding in reinforcement learning research, that despite great effort and the development of many new algorithms, few surpass the original temporal difference methods from Sutton and Barto (1998).

Shortcomings in the price markup methodology and competition levels have been identified and possible solutions proposed. The implications of increased computational expense for further development of this work have also been noted. AC optimal power flow adds enormously to simulation times when analysing an entire year of hourly trading interactions. The addition of bus voltage data was found to improve performance of the policy gradient methods, but it has not been

show if the same could not be achieved by perhaps using bus voltage angles from a DC optimal power flow formulation.

Chapter 7

Conclusions and Further Work

This final chapter summarises the conclusions that can be drawn from the results presented in this thesis and gives some ideas for further development of the contributions that have been made.

7.1 Summary Conclusions

This thesis has introduced the use of policy gradient reinforcement learning algorithms in modelling strategies of electricity market participants. Over the last two decades, competitive markets have become an essential component of electricity supply industries in many large countries. They will play an important role in the future as the world population continues to grow and finite primary energy fuel resources become increasingly scarce. Electric energy trade requires a unique market design, but radical architecture changes can not be experimented with on real systems.

Computational simulation is a well established technique for evaluating market design concepts and agent-based simulation is an approach that allows large complex systems to be modelled. There are many examples of learning algorithms being used to model electricity market participants in the literature, but policy gradient methods have not been previously applied. These methods use function approximation techniques to operate in environments with state and actions spaces that are continuous, discrete or mixed and have been successfully applied in robot control and other problems.

To examine the properties of policy gradient methods and compare their performance with previously applied value function based methods, a modular simulation framework has been defined and implemented. The framework uses a power exchange auction market model with nodal marginal pricing to provide an

environment in which agents can *learn to trade power* competitively.

The framework has first been used in a simulation that compares the convergence to Nash equilibria of four different learning algorithms. The simulation reproduced the findings of Krause et al. (2006) and presented similar results for policy gradient methods. They were found: to exhibit very different characteristics to value function based methods, to require a larger number of interactions before learning an optimal policy and to require low learning rate and exploration rate decay parameters for complex equilibria to be approached.

In a second simulation the framework was used to compare the same algorithms in a complex dynamic electricity trading environment. A reference electric power system model, designed for reliability analysis, was used to provide a realistic environment of a reasonable size. The algorithms were compared in their ability to observe and exploit constraints in the system as loads followed an hourly profile. It was not found that policy gradient methods could use additional bus voltage data to achieve greater performance than a action-value function based method or a stateful variant of the Roth-Erev technique. However, they were shown to learn effective strategies under noisy dynamic conditions and to improve performance using complex continuous state representations.

In conclusion, policy gradient methods are a valid option for modelling the strategies of electricity market participants. They can use profit feedback from an electricity market model to adjust the parameters of a policy function approximator in the direction of increased reward. Function approximators allow market participants to be modelled using agents that accept complex power system state data and produce offers for diverse portfolios of generators with price markups applied and capacity withheld. This thesis has compared a selection of techniques for participant modelling and provides guidance for algorithm and parameter selection.

7.2   Further Work

This final section highlights some of the shortcomings of the methodology presented in this thesis and explains how the models could be further developed. It introduces some alternative learning algorithms that might also be used to simulate electricity market participant behaviour. It explains is how a model formulated using data from National Grid Ltd. could be used in practical simulations of the UK electricity market and describes some other possibilities for using AC optimal power flow in agent-based electric power market simulation.

7.2.1 Parameter Sensitivity and Delayed Reward

The simulations presented in this thesis use typical algorithm parameters that are either the default values from PyBrain or inspired by the literature. No study of parameter sensitivity is performed. Alternative function approximation and back-propagation techniques could also be investigated. Parameter sensitivity analysis is typically conducted by the algorithm developers using standard benchmark problems (such as mazes or pole balancing problems) that are familiar to researchers in Artificial Intelligence and allow results to be compared. The shortage of published results and lack of standardised electricity trading models might limit the benefits of using this problem for general parameter sensitivity analysis.

The reward signals received by agents in all of the simulations presented in this thesis result directly from the agent's previous action. In reality, a market settlement process would introduce delays to payments for electricity production. Time did not permit value function based methods with eligibility traces (See Section 2.4.1) to be compared with policy gradient methods, but the ability to learn under delayed reward is a fundamental part of both reinforcement learning and market trade and deserves investigation in this context.

7.2.2 Alternative Learning Algorithms

This thesis has concentrated on traditional value function based methods, the Roth-Erev technique and two policy gradient reinforcement learning methods. However, there are other learning algorithms that have been published recently that could also be used in electric power trade simulations.

Riedmiller (2005) presented Neuro-Fitted Q-Iteration (NFQ) algorithms that attempt to overcome many of the problems experienced when implementing Q-learning methods with value function approximation using neural networks. They store all transition experiences and perform off-line updates using supervised learning techniques such as RProp (Riedmiller & Braun, 1993). The method has been shown to be robust against parameterisation and to learn quickly in standard benchmark tests and real-world applications (Kietzmann & Riedmiller, 2009).

The GQ(λ) algorithm by Maei and Sutton (2010) is another extension of Q-learning for operation in continuous environments. Convergence guarantees have been shown and the scaling properties suggest the method is suitable for large-scale reinforcement learning applications. A software implementation of GQ(λ)

has been developed by the authors and recently made available as open source.

Four new Natural Actor-Critic algorithms have been presented by Bhatnagar, Sutton, Ghavamzadeh, and Lee (2009). Like ENAC, they use function approximation techniques and are suitable for large-scale applications of reinforcement learning. Three of the algorithms are extensions to ENAC, but are fully incremental: the gradient computation is never reset while the policy is updated at every simulation step. The authors state a need to assess the ultimate utility of these algorithms through application in real-world problems.

This thesis provides a framework that would allow implementations of these algorithms to be assessed and used to research aspects of electricity markets.

7.2.3 UK Transmission System

Some of the more ambitious agent-based electricity market simulations have used stylised models of national transmission systems (Rastegar et al., 2009; Weidlich & Veit, 2006). This work has often been motivated by recent or expected changes to the arrangements in the associated regions. In the UK, nine large power stations are due to be decommissioned by 2016 in accordance with EU Large Combustion Plant Directive (National Electricity Transmission System Operator, 2007). Coupled with obligations, made in the Climate Change Act 2008, to cut greenhouse gas emissions by 80% of 1990 levels by 2050, coming years are likely to see major changes in the way the UK power system is operated. Examination of the situation could be enhanced by advanced participant behavioural models and accurate electric power system simulations such as those presented in this thesis.

Figure 7.1 illustrates a model of the UK transmission system that has been formulated from data provided by the National Electricity Transmission System Operator (2010). This model has been converted into PSS/E raw file format and is included with the code developed for this thesis (See Appendix A.11). It is currently too computationally expensive to be solved repeatedly in an agent-based simulation, but optimisation efforts might allow for it to be used in studies pertinent to the UK energy industry.

7.2.4 AC Optimal Power Flow

This thesis presents the first application of AC optimal power flow in electricity market simulation using reinforcement learning agents. AC optimal power flow formulations are more difficult to implement and more computationally expensive

88
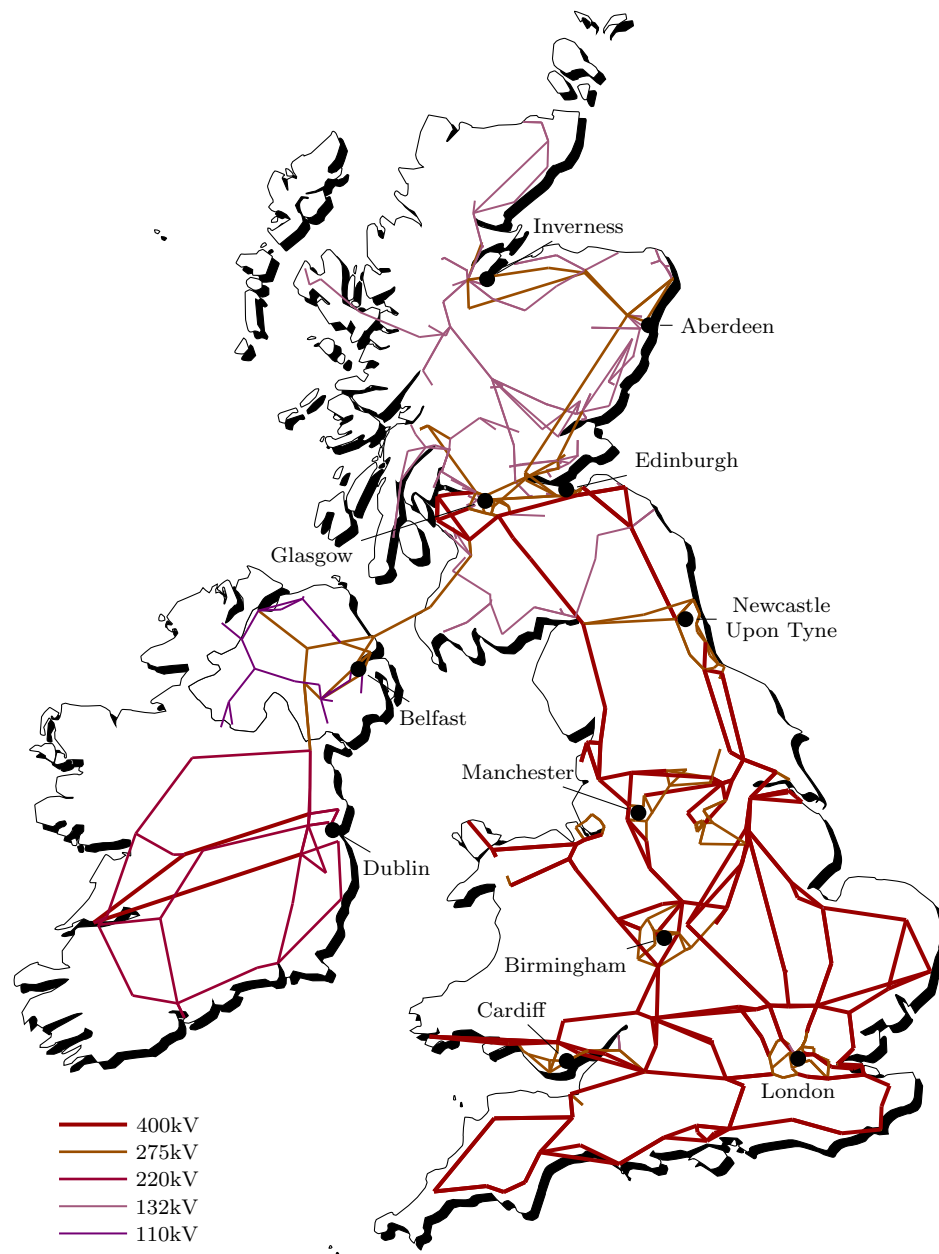
Figure 7.1: UK transmission system.

to solve than their linearised DC counterparts. The additional time and effort required for their use does not always add sufficient value to simulations. However, the option to use AC formulations does provide certain opportunities for further work.

The inclusion of reactive power costs in the objective function of an AC optimal power flow problem provides an opportunity to run auctions for voltage support in parallel with those for active power. These could be open to agents associated with reactive compensation equipment such as that commonly needed for wind farm developments. Traditionally, reactive power markets have been mostly of academic interest, but as the UK makes greater use of on and off-shore wind power, the topic could become of increasing interest and importance.

Bus voltages are not all assumed to be 1 per-unit in AC optimal power flow problems, but are part of the vector of optimisation variables. Adjusting phase shift angles, θ_{ph}, can offer a degree of control over the direction of power flows. The control of transformer tap ratios, τ, and phase shift angles by learning agents could become a topic of interest in congestion management research.

7.2.5 Multi-Market Simulation

Finally, the global economy is a holistic system of systems and the analysis of markets independently must be of limited value. Recent agent-based electricity market studies have investigated the interaction between electricity, gas and emission allowance markets (Kienzle et al., 2007; J. Wang et al., 2009).

Data for the UK gas transmission network provided by the National Electricity Transmission System Operator (2010) is of limited detail, compared to that for the electricity transmission system, but suitable models could be used to study the the relationships between UK gas and electricity markets. As in Kienzle et al. (2007), actions in the gas market would constrain the generators options to sell power in subsequent electricity auctions. Add to this the option to trade in emissions allowance markets and agents would be presented with large state and action spaces and would require suitably advanced learning methods.

Bibliography

Alam, M. S., Bala, B. K., Huo, A. M. Z., & Matin, M. A. (1991). A model for the quality of life as a function of electrical energy consumption. Energy, 16(4), 739-745.

Aleksandrov, V., Sysoyev, V., & Shemeneva, V. (1968). Stochastic optimization. Engineering Cybernetics, 5, 11-16.

Amerongen, R. van. (1989, May). A general-purpose version of the fast decoupled load flow. Power Systems, IEEE Transactions on, 4(2), 760-770.

Application of Probability Methods Subcommittee. (1979, November). IEEE reliability test system. Power Apparatus and Systems, IEEE Transactions on, PAS-98(6), 2047-2054.

Auer, P., Cesa-Bianchi, N., Freund, Y., & Schapire, R. E. (2003). The non-stochastic multiarmed bandit problem. SIAM Journal of Computing, 32(1), 48-77.

Baird, L. (1995). Residual algorithms: Reinforcement learning with function approximation. In Proceedings of the Twelfth International Conference on Machine Learning (p. 30-37). Morgan Kaufmann.

Bellman, R. E. (1961). Adaptive control processes – A guided tour. Princeton, New Jersey, U.S.A.: Princeton University Press.

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

Bhatnagar, S., Sutton, R. S., Ghavamzadeh, M., & Lee, M. (2009). Natural actor-critic algorithms. Automatica, 45(11), 2471–2482.

Bishop, C. M. (1996). Neural networks for pattern recognition (1st ed.). Oxford University Press, USA. Paperback.

Bower, J., & Bunn, D. (2001, March). Experimental analysis of the efficiency of uniform-price versus discriminatory auctions in the england and wales electricity market. Journal of Economic Dynamics and Control, 25(3-4), 561-592.

Bower, J., Bunn, D. W., & Wattendrup, C. (2001). A model-based analysis of

strategic consolidation in the german electricity industry. Energy Policy, 29(12), 987-1005.

Boyd, S., & Vandenberghe, L. (2004). Convex optimization. Cambridge University Press. Hardcover.

Bunn, D., & Martoccia, M. (2005). Unilateral and collusive market power in the electricity pool of England and Wales. Energy Economics.

Bunn, D. W., & Oliveira, F. S. (2003). Evaluating individual market power in electricity markets via agent-based simulation. Annals of Operations Research, 57-77.

Carpentier, J. (1962, August). Contribution à l'étude du Dispatching Economique. Bulletin de la Society Francaise Electriciens, 3(8), 431-447.

Cole, S. (2010, February 4). MatDyn [Computer software manual]. Katholieke Universiteit Leuven.

Department of Energy and Climate Change. (2009). Digest of United Kingdom Energy Statistics 2009. In (chap. 5). National Statistics – Crown.

Erev, I., & Roth, A. E. (1998). Predicting how people play games: Reinforcement learning in experimental games with unique, mixed strategy equilibria. The American Economic Review, 88(4), 848-881.

Ernst, D., Minoia, A., & Ilic, M. (2004, June). Market dynamics driven by the decision-making of both power producers and transmission owners. In Power Engineering Society General Meeting, 2004. IEEE (p. 255-260).

Fausett, L. (Ed.). (1994). Fundamentals of neural networks: architectures, algorithms, and applications. Upper Saddle River, NJ, USA: Prentice-Hall, Inc.

Gieseler, C. (2005). A Java reinforcement learning module for the Repast toolkit: Facilitating study and implementation with reinforcement learning in social science multi-agent simulations. Unpublished master's thesis, Department of Computer Science, Iowa State University.

Glimn, A. F., & Stagg, G. W. (1957, April). Automatic calculation of load flows. Power Apparatus and Systems, Part III. Transactions of the American Institute of Electrical Engineers, 76(3), 817-825.

Glynn, P. W. (1987). Likelilood ratio gradient estimation: an overview. In WSC '87: Proceedings of the 19th conference on winter simulation (p. 366-375). New York, NY, USA: ACM.

Goldfarb, D., & Idnani, A. (1983). A numerically stable dual method for solving strictly convex quadratic programs. Mathematical Programming, 27, 1-33.

Gordon, G. (1995). Stable function approximation in dynamic programming. In

Proceedings of the Twelfth International Conference on Machine Learning (p. 261-268). Morgan Kaufmann.

Grainger, J., & Stevenson, W. (1994). Power system analysis. New York: McGraw-Hill.

Guo, M., Liu, Y., & Malec, J. (2004, October). A new Q-learning algorithm based on the metropolis criterion. Systems, Man, and Cybernetics, Part B: Cybernetics, IEEE Transactions on, 34(5), 2140-2143.

ICF Consulting. (2003, August). The economic cost of the blackout: An issue paper on the northeastern blackout. (Unpublished)

IEEE Working Group. (1973, November). Common format for exchange of solved load flow data. Power Apparatus and Systems, IEEE Transactions on, 92(6), 1916-1925.

Kallrath, J., Pardalos, P., Rebennack, S., & Scheidt, M. (2009). Optimization in the energy industry. Springer.

Kienzle, F., Krause, T., Egli, K., Geidl, M., & Andersson, G. (2007, September). Analysis of strategic behaviour in combined electricity and gas markets using agent-based computational economics. In 1st European workshop on energy market modelling using agent-based computational economics (p. 121-141). Karlsruhe, Germany.

Kietzmann, T. C., & Riedmiller, M. (2009). The neuro slot car racer: Reinforcement learning in a real world setting. Machine Learning and Applications, Fourth International Conference on, 0, 311-316.

Kirschen, D. S., & Strbac, G. (2004). Fundamentals of power system economics. Chichester: John Wiley & Sons.

Krause, T., & Andersson, G. (2006). Evaluating congestion management schemes in liberalized electricity markets using an agent-based simulator. In Power Engineering Society General Meeting, 2006. IEEE.

Krause, T., Andersson, G., Ernst, D., Beck, E., Cherkaoui, R., & Germond, A. (2004). Nash equilibria and reinforcement learning for active decision maker modelling in power markets. In Proceedings of 6th IAEE European Conference 2004, modelling in energy economics and policy.

Krause, T., Beck, E. V., Cherkaoui, R., Germond, A., Andersson, G., & Ernst, D. (2006). A comparison of Nash equilibria analysis and agent-based modelling for power markets. International Journal of Electrical Power & Energy Systems, 28(9), 599-607.

Li, H., & Tesfatsion, L. (2009a, July). The ames wholesale power market test bed: A computational laboratory for research, teaching, and training. In

IEEE Proceedings, Power and Energy Society General Meeting. Alberta, Canada.

Li, H., & Tesfatsion, L. (2009b, March). Capacity withholding in restructured wholesale power markets: An agent-based test bed study. In Power systems conference and exposition, 2009 (p. 1-11).

Lincoln, R., Galloway, S., & Burt, G. (2007, May 23-25). Unit commitment and system stability under increased penetration of distributed generation. In Proceedings of the 4th International Conference on the European Energy Market, 2007. EEM 2007. Cracow, Poland.

Lincoln, R., Galloway, S., & Burt, G. (2009, May). Open source, agent-based energy market simulation with Python. In Proceedings of the 6th International Conference on the European Energy Market, 2009. EEM 2009. (p. 1-5).

Lincoln, R., Galloway, S., Burt, G., & McDonald, J. (2006, 6-8). Agent-based simulation of short-term energy markets for highly distributed power systems. In Proceedings of the 41st International Universities Power Engineering Conference, 2006. UPEC '06. (Vol. 1, p. 198-202).

Maei, H. R., & Sutton, R. S. (2010). Gq(λ): A general gradient algorithm for temporal-difference prediction learning with eligibility traces. In Proceedings of the third conference on artificial general intelligence. Lugano, Switzerland.

McCulloch, W., & Pitts, W. (1943, December 21). A logical calculus of the ideas immanent in nervous activity. Bulletin of Mathematical Biology, 5(4), 115-133.

Micola, A. R., Banal-Estañol, A., & Bunn, D. W. (2008, August). Incentives and coordination in vertically related energy markets. Journal of Economic Behavior & Organization, 67(2), 381-393.

Micola, A. R., & Bunn, D. W. (2008). Crossholdings, concentration and information in capacity-constrained sealed bid-offer auctions. Journal of Economic Behavior & Organization, 66(3-4), 748-766.

Minkel, J. R. (2008, August 13). The 2003 northeast blackout–five years later. Scientific American.

Momoh, J., Adapa, R., & El-Hawary, M. (1999, Feb). A review of selected optimal power flow literature to 1993. I. Nonlinear and quadratic programming approaches. Power Systems, IEEE Transactions on, 14(1), 96-104.

Momoh, J., El-Hawary, M., & Adapa, R. (1999, Feb). A review of selected optimal power flow literature to 1993. II. Newton, linear programming and interior point methods. Power Systems, IEEE Transactions on, 14(1), 105-111.

Moody, J., & Saffell, M. (2001, July). Learning to trade via direct reinforcement. IEEE Transactions on Neural Networks, 12(4), 875-889.

Moody, J., Wu, L., Liao, Y., & Saffell, M. (1998). Performance functions and reinforcement learning for trading systems and protfolios. Journal of Forecasting, 17, 441-470.

Naghibi-Sistani, M., Akbarzadeh-Tootoonchi, M., Javidi-D.B., M., & Rajabi-Mashhadi, H. (2006, November). Q-adjusted annealing for Q-learning of bid selection in market-based multisource power systems. Generation, Transmission and Distribution, IEE Proceedings, 153(6), 653-660.

Nash, J. F. (1950, January). Equilibrium points in n-person games. Proceedings of the National Academy of Sciences of the United States of America, 36(1), 48-49.

Nash, J. F. (1951, September). Non-cooperative games. The Annals of Mathematics, 54(2), 286-295. Available from http://dx.doi.org/10 .2307/1969529

National Electricity Transmission System Operator. (2007, September). Large combustion plant directive (Tech. Rep.). National Grid Electricity Transmission plc. (GCRP 07/32)

National Electricity Transmission System Operator. (2010, May). 2010 National Electricity Transmission System Seven Year Statement (Tech. Rep.). National Grid Electricity Transmission plc.

Nicolaisen, J., Petrov, V., & Tesfatsion, L. (2002, August). Market power and efficiency in a computational electricity market with discriminatory double-auction pricing. Evolutionary Computation, IEEE Transactions on, 5(5), 504-523.

Nicolaisen, J., Smith, M., Petrov, V., & Tesfatsion, L. (2000). Concentration and capacity effects on electricity market power. In Evolutionary Computation. Proceedings of the 2000 Congress on (Vol. 2, p. 1041-1047).

Overbye, T., Cheng, X., & Sun, Y. (2004, Jan.). A comparison of the AC and DC power flow models for LMP calculations. In System sciences, 2004. Proceedings of the 37th annual Hawaii international conference on (p. 9-).

Peshkin, L., & Savova, V. (2002). Reinforcement learning for adaptive routing. In Neural Networks, 2002. IJCNN 2002. Proceedings of the 2002 International Joint Conference on (Vol. 2, p. 1825-1830).

Peters, J. (2010). Policy gradient methods. (Available online: www.scholarpedia.org)

Peters, J., & Schaal, S. (2006, October). Policy gradient methods for robotics. In

Intelligent Robots and Systems, 2006 IEEE/RSJ International Conference on (p. 2219-2225).

Peters, J., & Schaal, S. (2008). Natural actor-critic. Neurocomputing, 71(7-9), 1180-1190.

Rastegar, M. A., Guerci, E., & Cincotti, S. (2009, May). Agent-based model of the Italian wholesale electricity market. In Energy Market, 2009. 6th International Conference on the European (p. 1-7).

Riedmiller, M. (2005). Neural fitted Q iteration - first experiences with a data efficient neural reinforcement learning method. In In 16th European conference on machine learning (pp. 317–328). Springer.

Riedmiller, M., & Braun, H. (1993). A direct adaptive method for faster backpropagation learning: the rprop algorithm.

Robbins, H. (1952). Some aspects of the sequential design of experiments. Bulletin American Mathematical Society, 58(5), 527-535.

Roth, A. E., Erev, I., Fudenberg, D., Kagel, J., Emilie, J., & Xing, R. X. (1995). Learning in extensive-form games: Experimental data and simple dynamic models in the intermediate term. Games and Economic Behavior, 8(1), 164-212.

Schaul, T., Bayer, J., Wierstra, D., Sun, Y., Felder, M., Sehnke, F., et al. (2010). PyBrain. Journal of Machine Learning Research, 11, 743-746.

Schweppe, F., Caramanis, M., Tabors, R., & Bohn, R. (1988). Spot pricing of electricity. Dordrecht: Kluwer Academic Publishers Group.

Sharpe, W. F. (1966, January). Mutual fund performance. Journal of Business, 119-138.

Sharpe, W. F. (1994). The Sharpe ratio. The Journal of Portfolio Management, 49-58.

Stott, B., & Alsac, O. (1974, May). Fast decoupled load flow. Power Apparatus and Systems, IEEE Transactions on, 93(3), 859-869.

Sun, J., & Tesfatsion, L. (2007a). Dynamic testing of wholesale power market designs: An open-source agent-based framework. Computational Economics, 30(3), 291-327.

Sun, J., & Tesfatsion, L. (2007b, June). Open-source software for power industry research, teaching, and training: A DC-OPF illustration. In Power Engineering Society General Meeting, 2007. IEEE (p. 1-6).

Sutton, R. S., & Barto, A. G. (1998). Reinforcement learning: An introduction. MIT Press. Gebundene Ausgabe.

Sutton, R. S., McAllester, D., Singh, S., & Mansour, Y. (2000). Policy gra-

dient methods for reinforcement learning with function approximation. In Advances in neural information processing systems (Vol. 12, p. 1057-1063).

Tellidou, A., & Bakirtzis, A. (2007, Novemeber). Agent-based analysis of capacity withholding and tacit collusion in electricity markets. Power Systems, IEEE Transactions on, 22(4), 1735-1742.

Tesauro, G. (1994). TD-Gammon, a self-teaching backgammon program, achieves master-level play. Neural Computation, 6(2), 215-219.

Tesfatsion, L., & Judd, K. L. (2006). Handbook of computational economics, volume 2: Agent-based computational economics (handbook of computational economics). Amsterdam, The Netherlands: North-Holland Publishing Co.

The International Energy Agency. (2010, Septemeber). Key world energy statistics 2010. Paris.

Tinney, W., & Hart, C. (1967, Novemeber). Power flow solution by Newton's method. Power Apparatus and Systems, IEEE Transactions on, 86(11), 1449-1460.

Tsitsiklis, J. N., & Roy, B. V. (1994). Feature-based methods for large scale dynamic programming. In Machine learning (p. 59-94).

United Nations. (2003, December 9). World population in 2300. In Proceedings of the United Nations, Expert Meeting on World Population in 2300.

U.S.-Canada Power System Outage Task Force. (2004, April). Final report on the august 14, 2003 blackout in the United States and Canada: Causes and recommendations (Tech. Rep.). North American Electric Reliability Corporation.

Veit, D., Weidlich, A., Yao, J., & Oren, S. (2006). Simulating the dynamics in two-settlement electricity markets via an agent-based approach. International Journal of Management Science and Engineering Management, 1(2), 83-97.

Vengerov, D. (2008). A gradient-based reinforcement learning approach to dynamic pricing in partially-observable environments. Future Generation Computer Systems, 24(7), 687-693.

Visudhiphan, P. (2003). An agent-based approach to modeling electricity spot markets. Unpublished doctoral dissertation, Department of Electrical Engineering and Computer Science, Massachusetts Institute of Technology.

Visudhiphan, P., & Ilic, M. (1999, February). Dynamic games-based modeling of electricity markets. In Power Engineering Society 1999 Winter Meeting, IEEE (Vol. 1, p. 274-281).

Wang, H., Murillo-Sanchez, C., Zimmerman, R., & Thomas, R. (2007, Aug.). On

computational issues of market-based optimal power flow. Power Systems, IEEE Transactions on, 22(3), 1185-1193.

Wang, J., Koritarov, V., & Kim, J.-H. (2009, July). An agent-based approach to modeling interactions between emission market and electricity market. In Power Energy Society General Meeting, 2009. PES 2009. IEEE (p. 1-8).

Weidlich, A., & Veit, D. (2006, July 7-10). Bidding in interrelated day-ahead electricity markets - insights from an agent-based simulation model. In Proceedings of the 29th IAEE International Conference.

Weidlich, A., & Veit, D. (2008, July). A critical survey of agent-based wholesale electricity market models. Energy Economics, 30(4), 1728-1759.

WG 31.04. (1983). Electric power transmission at voltages of 1000 kV and above plans for future AC and DC transmission. Electra. (ELT_091_3)

Williams, R. J. (1992). Simple statistical gradient-following algorithms for connectionist reinforcement learning. In Machine learning (p. 229-256).

Wood, A. J., & Wollenberg, B. F. (1996). Power Generation Operation and Control (second ed.). New York: Wiley, New York.

Yao, J., Adler, I., & Oren, S. S. (2008). Modeling and computing two-settlement oligopolistic equilibrium in a congested electricity network. Operations Research, 56(1), 34-47.

Yao, J., Oren, S. S., & Adler, I. (2007). Two-settlement electricity markets with price caps and cournot generation firms. European Journal of Operational Research, 181(3), 1279-1296.

Zimmerman, R. (2010, March 19). MATPOWER 4.0b2 User's Manual [Computer software manual]. School of Electrical Engineering, Cornell University, Ithaca, NY 14853.

Zimmerman, R., Murillo-Sánchez, C., & Thomas, R. J. (2009, July). MATPOWER's extensible optimal power flow architecture. In IEEE PES General Meeting. Calgary, Alberta, Canada.

Appendix A

Open Source Electric Power Engineering Software

For the purposes of this thesis the Matlab source code from MATPOWER was translated into the Python programming language and released as a project named Pylon[1] (Lincoln et al., 2009). It was translated to allow existing implementations of policy gradient reinforcement learning methods, from the PyBrain machine learning library (Schaul et al., 2010), to be coupled with MATPOWER's scalable and extensible optimal power flow formulations. With permission from the MATPOWER developers, the resulting code was released under the terms of the Apache License, version 2.0, and this section describes the project in the context of other open source Electrical Power Engineering software tools to illustrate the contribution made.

Table A.1 lists the programming language and license for each of the projects reviewed and shows which projects feature: power flow (PF), multi-phase power flow (MPF), DC optimal power flow (DCOPF), AC optimal power flow (ACOPF), continuation power flow (CPF), small-signal stability analysis (SSSA), time domain simulation (TDS), state estimation (SE), sparse matrices (SP), a graphical user interface (GUI) and reinforcement learning (RL) agent based simulation.

A.1 MATPOWER

Since 1996, a team of researchers from the Power Systems Engineering Research Center (PSERC) at Cornell University have been developing MATPOWER: a package of Matlab[2] workspace files for solving power flow and optimal power flow

---

[1]http://packages.python.org/Pylon/
[2]Matlab is a registered trademark of The Mathworks, Inc.

Package	Language	Licence	PF	MPF	DCOPF	ACOPF	CPF	SSSA	TDS	SE	SP	GUI	RL
AMES	Java	GPL			•							•	•
DCOPFJ	Java	GPL			•								
GridLab-D	C++	BSD	•	•							•		
MatDyn	Matlab								•		•		
MATPOWER	Matlab	GPL	•	•	•	•	•			•	•		
OpenDSS	Pascal	BSD	•								•		
PSAT	Matlab	GPL	•			•	•	•	•	•	•	•	•
Pylon	Python	Apache	•		•	•	•				•	•	
TEFTS	C		•				•		•		•	•	
VST	Matlab						•	•	•		•	•	
UWPFLOW	C						•				•		

Table A.1: Open source electric power engineering software feature matrix.

problems (Zimmerman, Murillo-Sánchez, & Thomas, 2009). Initial development was part of the PowerWeb project in which the team created a power exchange auction market simulator that could be accessed by multiple users simultaneously through a web-browser interface. MATPOWER was originally available under a custom license that permitted use for any purpose providing the project and authors were cited correctly, but since version 4.0b3 it has been released under the less permissive GNU General Public License (GPL), version 3. MATPOWER has become very popular in education and research and has an active mailing list that is moderated by Dr Ray Zimmerman of PSERC.

MATPOWER includes five solvers for AC and DC power flow. The default solver uses Newton's method (Tinney & Hart, 1967) with the full Jacobian matrix updated at each iteration. Two variations on the fast decoupled method (Stott & Alsac, 1974) described in Amerongen (1989) provide quicker convergence for certain networks. The standard Gauss-Seidel method (Glimn & Stagg, 1957) is provided for academic purposes and the DC solver provides non-iterative solutions. The properties of Matlab sparse matrices are exploited to allow solvers to scale well with very large systems. All functions are run from the Matlab command-line or from within users programs and no graphical user interface is provided.

Starting with version 4.0, MATPOWER includes the Matlab Interior Point Solver (MIPS) that can be used for solving DC and AC optimal power flow problems (H. Wang et al., 2007). Previously, FMINCON from the Matlab Optimization Toolbox[3] was required or one of a suite of high performance closed-source solvers: TSPOPF is a collection of three AC optimal power flow solvers, implemented in C and released as Matlab MEX files. It includes the original implementation of the step-controlled interior point method from which MIPS was derived. MINOPF provides an interface to the Fortran based MINOS[4] solver, developed at the Systems Optimization Laboratory at Stanford University, and is available only for educational and research purposes. Since version 4.0b4 MATPOWER has also included an interface to IPOPT from the COIN-OR project that provides an alternative open source solution to MIPS. DC optimal power flow problems can be solved with a Quadratic Programming interface to MIPS or using a MEX interface to BPMPD: a commercial interior point method for linear and quadratic programming.

MATPOWER has an extensible optimal power flow formulation that allows users to introduce additional optimisation variables and problem constraints. It

---

[3]Optimization Toolbox is a registered trademark of The Mathworks, Inc.
[4]MINOS is trademark of Stanford Business Software, Inc.

is used internally to extend the standard DC and AC formulations to support piecewise linear cost functions, dispatchable loads, generator PQ capability curves and branch angle difference limit constraints. Examples of possible additional extensions include: reserve requirements, environmental costs and contingency constraints.

MATPOWER currently runs on Matlab, a commercial software product from The Mathworks that is supported on all major platforms, or on GNU/Octave, a free program for numerical computation with strong Matlab compatibility.

A.2 MATDYN

MATDYN is an extension to MATPOWER developed by Stijn Cole from the Katholieke Universiteit Leuven for dynamic analysis of electric power systems. It was first released in 2009 under MATPOWER's custom license. It uses the same programming style and extends the MATPOWER case format with structs for dynamic generator and event data. MATDYN uses MATPOWER to obtain a power flow solution that is then used in solving a system of differential algebraic equations representing the power system. Results from MATDYN have been validated by Cole (2010) against those obtained from PSS/E[5] and the Power System Analysis Toolbox (See Section A.3, below) and show good correspondence.

A.3 PSAT

The Power System Analysis Toolbox (PSAT) is a Matlab toolbox for static and dynamic analysis of electric power systems developed by Federico Milano of the University of Castilla. It is released under the terms of the GNU GPL version 2 and offers routines for:

- Power flow,

- Bifurcation analysis,

- Optimal power flow,

- Small signal stability analysis,

- Time domain simulation and

---

[5]PSS/E is a registered trademark of Siemens Power Transmission & Distribution, Inc. Power Technologies International.

- Phasor measurement unit placement.

A large number of input data formats are supported through Perl scripts and simulation reports can be exported as plain text, Excel spreadsheets or LaTeX 2_ε code. PSAT may be run from the Matlab command-line or through a Matlab based graphical user interface. The interface can be used with Simulink[6] to construct cases such as the UK Generic Distribution System network shown in Figure A.1. A slightly modified version of PSAT that can be run from the Gnu/Octave command-line is also available.

---

[6]Simulink is a registered trademark of The Mathworks, Inc.
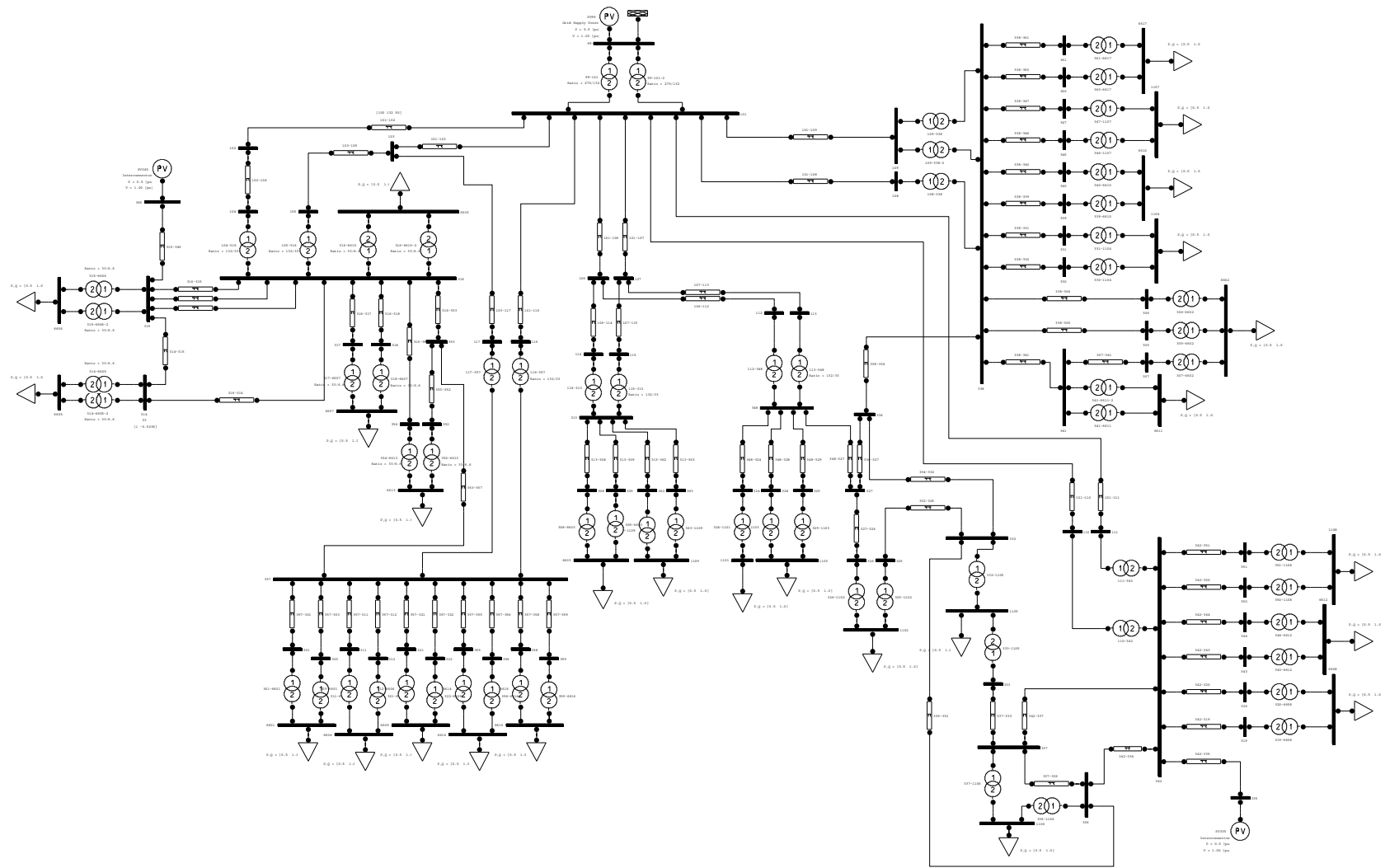
Figure A.1: UKGDS EHV3 model in PSAT Simulink network editor.

Optimal power flow problems are solved via an interface to the General Algebraic Modeling System (GAMS). GAMS defines optimisation problems using a high-level modelling language and has a large solver portfolio that includes all of the major commercial and academic solvers. The interface can be used for solving single period optimal power flow problems where the objective function can model maximisation of social benefit, maximisation of the distance to the maximum loading condition or a multi-objective combination of these. Multi-period optimal power flow is formulated as a mixed integer problem with linearised power balance constraints. The objective function models maximisation of social welfare, but is extended to include start-up and shutdown costs.

Power flow and dynamic data are often separated in electric power simulation tools, but in PSAT they are integrated. This combined with the large number of routines supported by PSAT can make the code base difficult to understand and modify. However, comprehensive documentation is included with PSAT and the mailing list is very active. The price of GAMS licenses and the need for optimal power flow problems to be converted to the GAMS language before being solved may be considered barriers to its selection for certain projects.

A.4   UWPFLOW

UWPFLOW is a research tool for voltage stability analysis developed at the University of Waterloo, Ontario, and the University of Wisconsin-Madison. It is written in ANSI-C and is available as open source for research purposes only. The program can be run with the terminal command

```
$ uwpflow [-options] input_file
```

where `input_file` is the path to a data file in the IEEE common data format (CDF) (IEEE Working Group, 1973), that may contain High-Voltage Direct Current (HVDC) and Flexible Alternating Current Transmission System (FACTS) device data. Output is also in the CDF and can include additional data for post-processing, including values for nose curve plots. An interface to UWPFLOW is provided with PSAT and can be used for bifurcation analysis.

A.5   TEFTS

The University of Waterloo also hosts TEFTS – a transient stability program for studying energy functions and voltage stability phenomena in AC/HVDC

dynamic power system models. It too is written in ANSI-C and is licensed for research purposes only. An executable file for DOS is provided and the source package contains a simple example.

A.6   VST

The Voltage Stability Toolbox (VST) is a Matlab toolbox, developed at the Center for Electric Power Engineering at Drexel University in Philadelphia, for investigating stability and bifurcation issues in power systems. The source is available for any purpose providing that the authors are suitably cited. VST features routines for:

- Power flow,

- Time domain simulation,

- Static and dynamic bifurcation analysis,

- Singularity analysis and

- Eigenvalue analysis.

The feature matrix in Table A.1 shows the similar capabilities of VST and PSAT. It was developed around the same time and has the same goals for educational and research applications. However, VST does not have the same quality of documentation, graphical user interface or such an active community of users and developers.

A.7   OpenDSS

In November 2008, the Open Distribution System Simulator (OpenDSS) was released by the Electric Power Research Institute (EPRI) as open source. Development of OpenDSS began in April 1997 and it has been used extensively in studies of distribution systems including distributed generation impact assessments.

OpenDSS supports steady-state analysis in the frequency domain, including power flow, harmonics and dynamics. Arbitrary n-phase unbalanced circuit analysis is supported using an object orientated data model. Circuit elements are defined in Object Pascal and solutions are obtained using KLUSolve: a linear sparse matrix solver written in C and C++and developed specifically for solving electrical circuits. The OpenDSS Pascal code is available under the Berkeley

Software Distribution (BSD) license, which allows use for almost any purpose. KLUSolve, is available under the Gnu Lesser GPL. Circuits are defined in scripts, using a domain specific language, that may be executed through a graphical user interface or a Common Object Model (COM) interface. The user interface also provides circuit data editing, plotting and power flow visualisation tools.

The power flow solver is fast and can be configured for repeated studies using daily, yearly or duty-cycle data. The multi-phase circuit model allows complex transformer models and fault conditions to be defined and three short-circuit analysis methods are provided. The heritage of OpenDSS is in harmonics and dynamics analysis and it does not support system optimisation.

A.8   GridLAB-D

GridLAB-D is an energy system simulation and analysis tool designed to investigate the latest energy technologies. The project was initiated by the U.S. Department of Energy in 2004 and developed at Pacific Northwest National Laboratory. It was released under a BSD-style license in September 2009 and has since been developed in collaboration with industry and academia.

A distributed simulation architecture is used to coordinate energy system component interactions over short and long timescales. The core of GridLAB-D is made up of modules for simulating: distribution and transmission systems, commercial and residential buildings, energy markets, power system faults and meteorological systems. GridLAB-D is written in C++ and uses a domain specific language to define models. Additional modules can be written in C++ or Java and Python is under consideration. It is designed for multicore/multiprocessor parallelism and the developers intend to use it simulate large areas of the U.S. on supercomputers. The source code includes reports and data from the Olympic Peninsula Project: a futuristic energy pricing experiment that provides a practical demonstration of GridLAB-D in operation.

GridLAB-D is a unique simulation tool that has the potential to play an important role in future energy system development. Its size and complexity can make for a steep learning curve, but extensive documentation is provided and training courses are run periodically. Activity on the mailing lists is low, suggesting poor uptake, but the software is actively supported and a new version is under development.

A.9 AMES

The AMES (Agent-based Modeling of Electricity Systems) power market testbed is a software package that models core features of the Wholesale Power Market Platform: a market design proposed by the Federal Energy Regulatory Commission (FERC) in April 2003 for common adoption in regions of the U.S. (Sun & Tesfatsion, 2007a). The market design features:

- A centralised structure managed by an independent market operator,

- Parallel day-ahead and real-time markets and

- Locational marginal pricing.

Learning agents represent load serving entities or generating companies and learn using Roth-Erev reinforcement learning methods, implemented using the Repast agent simulation toolkit (Gieseler, 2005). Agents learn from the solutions of hourly bid/offer based DC-OPF problems formulated as quadratic programs using the DCOPFJ package (Sun & Tesfatsion, 2007b) (See Section A.10, below).

The capabilities of AMES are demonstrated using a 5-bus network model in Li and Tesfatsion (2009a). The model is provided with AMES and a step-by-step tutorial describes how it may be used. AMES comes with a Swing-based graphical user interface with plotting and table editor tools and is released under the GNU GPL, version 2.

A.10 DCOPFJ

To solve market problems defined in AMES, researchers at Iowa State University developed a stand-alone DC optimal power flow solver in Java named DCOPFJ. It formulates optimal power flow problems as convex quadratic programs which are solved using QuadProgJ. The same researcher developed QuadProgJ as an independent solver that uses a dual active set strictly convex quadratic programming algorithm (Goldfarb & Idnani, 1983). DCOPFJ requires generator costs to be modelled as polynomial functions, of second order or less, and does not exploit sparse matrix features.

A.11 PYLON

Pylon is a translation of MATPOWER v4.0b2 and MATDYN v1.2 to the Python programming language. It has extensions for agent-based electricity market sim-

ulation that provide features similar to those of AMES. Both the DC and AC formulations of the extensible optimal power flow model from Matpower are implemented (Zimmerman et al., 2009). Either a Python version of MIPS or an interface to IPOPT from COIN-OR can be used to compute solutions. The sparsity of the problems is exploited throughout the solution process using matrix packages from SciPy and bindings to SuperLU or UMFPACK for LU decomposition and solving sparse sets of linear equations. Scripts are provided for reading and writing data files in PSS/E, Matpower and PSAT format. A wide variety of learning methods are available in Pylon due to its use of the PyBrain machine learning library (Schaul et al., 2010). PyBrain also provides the artificial neural network models used for policy function approximation, that may be accelerated using C extension modules from the ARAC sub-project.

In addition to its market simulation capabilities, Pylon also features solvers for power flow problems (using fast decoupled or Newton's method), state estimation, continuation power flow and time domain simulation. Pylon includes both a text interface and a graphical user interface (GUI) based on TkInter: which is included with Python and imposes no additional dependencies. A feature rich GUI is provided by plug-ins for Puddle: an extensible, GUI toolkit independent integrated development environment, developed for the purposes of this thesis also.

The use of matrix libraries from NumPy and SciPy has allowed Pylon (with the permission of the Matpower developers) to be released under the Apache license, version 2.0. This allows Pylon to be used as a library in proprietary software as well as free and open source tools since derivatives of the source code may be made available under more restrictive terms than the original Apache license. This is in contrast to strong "copyleft" licenses, such as the Gnu GPL, that require the same rights to be preserved in modified versions of the work.

Summary

A diverse range of open source Electric Power Engineering tools are available. Implementations of most of the traditional power system analysis routines can be found and many offer performance comparable with proprietary offerings. Various programming languages are used, but Matlab is the most popular choice.

Several projects are licensed under the Gnu GPL and it ensures that all users have access to the full source code. This does impose restrictions on the redistribution of projects that use the routines and this can be a barrier to use in

certain types of project. To encourage commercial use and promote industrial involvement, two large code bases (OpenDSS and GridLAB-D) have recently been released under weak copyleft licenses. Pylon was developed using specific scientific computing libraries and permission was obtained from the developers of MATPOWER to allow release under a similarly permissive license. Most of the projects described above are led and developed by one individual and contributions from the user community are typically minimal. It is hoped that Pylon's use of a popular free programming language and its liberal licensing conditions will encourage community involvement and lead to inventive combinations of simulation routines and web technologies in the development of intelligent electric power grids.

Appendix B

Case Data

This appendix provides data for the electric power system models used in Chapters 5 and 6.

B.1   6-Bus Case

The data for the six bus case adapted from Wood and Wollenberg (1996, pp. 104, 112, 119, 123-124, 549) is presented in this section. The data was imported from the "case6ww.m" case file provided with MATPOWER. Table B.1 lists the bus data, Table B.2 lists the generator data and Table B.3 lists the branch data.

Table B.1: 6-bus case bus data.

Bus	P_d	Q_d	G_s	B_s	V_{base}	V_{max}	V_{min}
1	0	0	0	0	230	1.05	1.05
2	0	0	0	0	230	1.05	1.05
3	0	0	0	0	230	1.07	1.07
4	70	70	0	0	230	1.05	0.95
5	70	70	0	0	230	1.05	0.95
6	70	70	0	0	230	1.05	0.95

Table B.2: 6-bus case generator data.

Bus	P_{max}	P_{min}	V_g	Q_{max}	Q_{min}
1	1.05	200	50	100	-100
2	1.05	150	37.5	100	-100
3	1.07	180	45	100	-100

Table B.3: 6-bus case branch data.

From	To	r	x	b_c	S_{max}	τ	θ_{ph}
1	2	0.1	0.2	0.04	40	0	0
1	4	0.05	0.2	0.04	60	0	0
1	5	0.08	0.3	0.06	40	0	0
2	3	0.05	0.25	0.06	40	0	0
2	4	0.05	0.1	0.02	60	0	0
2	5	0.1	0.3	0.04	30	0	0
2	6	0.07	0.2	0.05	90	0	0
3	5	0.12	0.26	0.05	70	0	0
3	6	0.02	0.1	0.02	80	0	0
4	5	0.2	0.4	0.08	20	0	0
5	6	0.1	0.3	0.06	40	0	0

B.2   IEEE Reliability Test System

This section provides data for the modified IEEE Reliability Test System that was imported from the "case24_ieee_rts.m" case file, provided with MATPOWER and was originally contributed by Bruce Wollenberg. Table B.4 lists the bus data, Table B.5 lists the generator data, Table B.6 lists the branch data and Table **??** lists the generator cost data provided by Georgia Tech Power Systems Control and Automation Laboratory.

Table B.4: IEEE RTS bus data.

Bus	P_d	Q_d	G_s	B_s	V_{base}	V_{max}	V_{min}
1	108	22	0	0	138	1.05	0.95
2	97	20	0	0	138	1.05	0.95
3	180	37	0	0	138	1.05	0.95
4	74	15	0	0	138	1.05	0.95
5	71	14	0	0	138	1.05	0.95
6	136	28	0	-100	138	1.05	0.95
7	125	25	0	0	138	1.05	0.95
8	171	35	0	0	138	1.05	0.95
9	175	36	0	0	138	1.05	0.95
10	195	40	0	0	138	1.05	0.95
11	0	0	0	0	230	1.05	0.95
12	0	0	0	0	230	1.05	0.95
13	265	54	0	0	230	1.05	0.95
14	194	39	0	0	230	1.05	0.95
15	317	64	0	0	230	1.05	0.95
16	100	20	0	0	230	1.05	0.95
17	0	0	0	0	230	1.05	0.95
18	333	68	0	0	230	1.05	0.95
19	181	37	0	0	230	1.05	0.95
20	128	26	0	0	230	1.05	0.95
21	0	0	0	0	230	1.05	0.95
22	0	0	0	0	230	1.05	0.95
23	0	0	0	0	230	1.05	0.95
24	0	0	0	0	230	1.05	0.95

Table B.5: IEEE RTS generator data.

Bus	P_{max}	P_{min}	V_g	Q_{max}	Q_{min}	Type
1	152	30.4	1.035	60	-50	U76
2	152	30.4	1.035	60	-50	U76
7	300	75	1.025	180	0	U100
13	591	207	1.02	240	0	U197
14	0	0	0.98	200	-50	SynCond
15	155	54.3	1.014	80	-50	U155
16	155	54.3	1.017	80	-50	U155
18	400	100	1.05	200	-50	U400
21	400	100	1.05	200	-50	U400
22	300	60	1.05	96	-60	U50
23	310	108.6	1.05	160	-100	U155
23	350	140	1.05	150	-25	U350

Table B.6: IEEE RTS branch data.

From	To	r	x	b_c	S_{max}	τ	θ_{ph}
1	2	0.0026	0.0139	0.4611	175	0	0
1	3	0.0546	0.2112	0.0572	175	0	0
1	5	0.0218	0.0845	0.0229	175	0	0
2	4	0.0328	0.1267	0.0343	175	0	0
2	6	0.0497	0.192	0.052	175	0	0
3	9	0.0308	0.119	0.0322	175	0	0
3	24	0.0023	0.0839	0	400	1.03	0
4	9	0.0268	0.1037	0.0281	175	0	0
5	10	0.0228	0.0883	0.0239	175	0	0
6	10	0.0139	0.0605	2.459	175	0	0
7	8	0.0159	0.0614	0.0166	175	0	0
8	9	0.0427	0.1651	0.0447	175	0	0
8	10	0.0427	0.1651	0.0447	175	0	0
9	11	0.0023	0.0839	0	400	1.03	0
9	12	0.0023	0.0839	0	400	1.03	0
10	11	0.0023	0.0839	0	400	1.02	0
10	12	0.0023	0.0839	0	400	1.02	0
11	13	0.0061	0.0476	0.0999	500	0	0
11	14	0.0054	0.0418	0.0879	500	0	0
12	13	0.0061	0.0476	0.0999	500	0	0
12	23	0.0124	0.0966	0.203	500	0	0
13	23	0.0111	0.0865	0.1818	500	0	0
14	16	0.005	0.0389	0.0818	500	0	0
15	16	0.0022	0.0173	0.0364	500	0	0
15	21	0.0063	0.049	0.103	500	0	0
15	21	0.0063	0.049	0.103	500	0	0
15	24	0.0067	0.0519	0.1091	500	0	0
16	17	0.0033	0.0259	0.0545	500	0	0
16	19	0.003	0.0231	0.0485	500	0	0
17	18	0.0018	0.0144	0.0303	500	0	0
17	22	0.0135	0.1053	0.2212	500	0	0
18	21	0.0033	0.0259	0.0545	500	0	0
18	21	0.0033	0.0259	0.0545	500	0	0
19	20	0.0051	0.0396	0.0833	500	0	0
19	20	0.0051	0.0396	0.0833	500	0	0
20	23	0.0028	0.0216	0.0455	500	0	0
20	23	0.0028	0.0216	0.0455	500	0	0
21	22	0.0087	0.0678	0.1424	500	0	0