Chapter 7
Control and Management of Active Buildings

Ben Wooding, Vahid Vahidinasab, Milad Kazemi, Sadegh Soudjani

7.1 Introduction to Control of Active Building Energy Systems

7.1.1 Control Approaches for Enabling Energy Services Provision from Buildings

The power network is becoming increasingly intermittent as the contribution from renewable energy generation rises. To maintain stability and functionality of the power network, storage of renewable energies and demand-side control techniques are required. *Smart grids* provide the communication infrastructure to accomplish this goal. Smart grid control originated from the idea that the demand-side of the power grid can shift or shed load to reduce the strain on the network, while also maintaining consumer satisfaction and other specialist requirements [1]. The main benefit of a *smart city*, is to help its citizens by making city-related decisions. A smart grid differs from a smart city since its communication network revolves around optimising the power network, while a smart city also considers other city-level features as well as energy provision. Both smart grid and smart city benefit from timely and relevant information transfer.

Ben Wooding
School of Computing, Newcastle University, 1 Science Square, Newcastle upon Tyne, United Kingdom, NE4 5TG e-mail: b.wooding1@newcastle.ac.uk

Vahid Vahidinasab
Department of Engineering, School of Science and Technology, Nottingham Trent University, Nottingham, United Kingdom, NG11 8NS e-mail: vahid.vahidinasab@ntu.ac.uk

Milad Kazemi
School of Computing, Newcastle University, 1 Science Square, Newcastle upon Tyne, United Kingdom, NE4 5TG e-mail: milad.kazemi2@newcastle.ac.uk

Sadegh Soudjani
School of Computing, Newcastle University, 1 Science Square, Newcastle upon Tyne, United Kingdom, NE4 5TG e-mail: sadegh.soudjani@newcastle.ac.uk

To give a supporting example, consider the energy management system of a country. The power usage of each house in a city is measured regularly, and an aggregate model can be created to show the energy demands of different cities. The flow of power to those cities is also calculated, and expected needs are considered. From this data, an energy model can be created, and the power system can be controlled so the needs of each city are met. The population of the city is usually proportional to the city's energy usage. Services such as hospitals, schools, factories and universities, are also more frequent for higher population numbers. City centres and industrial zones will use more resources than residential areas, so a control scheme needs to consider all of these aspects when being created. One control could be that higher use prices should be charged to areas that use more energy, with the intent of bringing their usage down [2]. We will look at these energy management concepts in greater depth in the following sections.

7.1.2 Pros and Cons of Control Approaches

Smart cities and smart grids are valuable as they allow society to make intelligent and efficient decisions, with the support of regularly updating information. Control is the central principle for a smart grid and is essential for safe and efficient operation. It is a broad term that covers various processes and system objectives.

At a district or city level, energy management requires incorporating the growth of the network, more energy generating resources and storage options, and increased decentralisation caused by devices becoming increasingly interconnected, e.g. over the internet. On top of this, the grid demand is flexible as devices can connect and disconnect at any moment, and there is also flexibility in the energy prices. All of these lead to additional control complexities [3].

One difficulty for control is the non-linearity of the system being controlled. Non-linear system optimisations require complex calculations and are time consuming. The required computations most often rely on models of the control system, but building these models with significant accuracy is another difficulty [4]. Available techniques to tackle these challenges include system identification [5], linearisation around a fixed point [6], and model-free data-driven control [7]. But this is still a significant challenge to overcome for most control systems.

Another potential challenge associated with smart grid control is security. Smart grid control revolves around two-way power and information flow of the communication networks. Should an attacker gain the correct privileges, they may be able to deliberately jeopardise grid stability with induced load surges [8], or other methods. There are several reasons this attack is unlikely, simply the more distributed and diverse the network is the more security protocols and devices that would need to be exploited. But the negative use of control schemes for destabilisation should be considered due to the potential large-scale damages if such incidents happen.

Smart infrastructures, such as smart buildings in a smart city, are increasing in prevalence. Smart infrastructure is self-monitoring, self-governing and able to communicate with other aspects of the smart grid. From this, three research topics have emerged surrounding their control. Firstly, consumers have been empowered to interact with the new resources available to them and systems with multiple decision-makers have emerged. Secondly, incentives are created to enable flexible consumption, such as cheaper prices when consuming electricity at night. This is known as *transactive control* [9]. Finally, resilient control is a term given to protecting the system from large system failures, and does so by leveraging the communication between the smart devices. These areas each support the higher level smart grid control, but also the end-users who receive incentives and negotiating powers in the energy market [10].

Moreover, control of the smart grid enables the increased integration of renewable energies. Renewable energy generation is uncertain and intermittent, but if properly controlled can be a big step forward to the global aim of net-zero carbon. Ultimately, all the positives and negatives presented in this section hinge on the quality of the control schemes. High quality control, from accurate system models and security defences, has no negatives of note. However, poor quality control from inaccurate models and flawed security, would be more harmful than helpful. Deciding which category a system falls into is a different challenge entirely and requires appropriate research to quantitatively characterise *quality* of a given control scheme applied to a system.

7.2 Coordination Structures for Management and Control of the Energy Systems

In this section, we introduce the coordinating structures available in smart grids. There are two key characteristics of such coordination frameworks, these are the *system structure* and the *energy resource type*. System structure considers the configuration of the power units included within the system and their interconnections. The energy resource type may be renewable or non-renewable energy, and this has influence over which coordination framework would optimise the system performance [11].

7.2.1 Coordinated Structures from the System Perspective

The possibility of large-scale energy transfer became a reality with the rise of the industrial revolution. The process involved extracting natural resources, transporting them and then converting the resource to energy via large turbine-generators. Even today, all three steps of the process are expensive and involve operations to a scale that only large corporations or bodies of government can handle. Delivering the energy to users was done through a *centralised framework*: a central large-scale

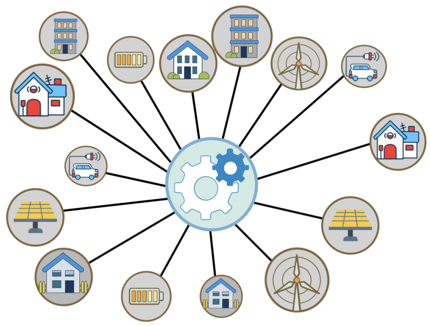

Fig. 7.1: Centralised energy systems scheme.

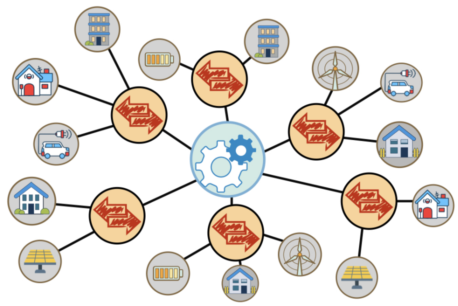

Fig. 7.2: Decentralised energy systems scheme.

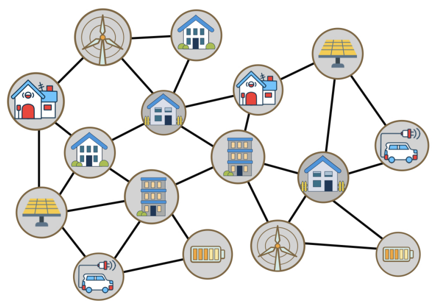

Fig. 7.3: Distributed energy systems scheme.

generation unit sending the energy to global users far from the original generator, see Fig. 7.1.

As technology advances, generation units have become more readily available and at a reasonable price. This has created an opportunity for smaller companies and individuals to share in the market. A *decentralised framework* is shown in Fig. 7.2, where small generation units provide energy to local users and communication channels are formed between generators to share excess energy or store it.

In the decentralised and centralised approaches, the local users only consume energy from the power network. However, it is becoming more common for those users to also take part in power generation. For example, a user may decide to install a *photovoltaic (PV) panel* at their house to generate some electricity and reduce energy bill costs. This contributed to the development of the *distributed framework*. In this framework, local users can become *prosumers*, producing and consuming energy, simultaneously. A prosumer could become energy self-sufficient, but by forming a distributed network with other prosumers, any excess energy can be shared. Distributed networks can then connect to share that energy wider, see

Fig. 7.3. As there is added complexity in the distributed framework, it is also more complex when designing control schemes compared with centralised controls [12].

The three schemes discussed above and shown in Figs. 7.1-7.3 could reflect possible energy flow between component of the network. They could also reflect possible communications between these components. Interpreted differently, a lack of link between two components means lack of direct energy transfer or communication between these two components. As such, all these constraints should be taken into account when designing a control scheme for the whole network.

7.2.2 Coordinated Structures from the Perspective of Energy Resources

Energy resources can be divided into two categories of renewable and non-renewable, and influence the choice of framework to be implemented. Extraction of non-renewable fossil fuels such as oil as well as the required infrastructure for their transportation are expensive. Therefore, a large-scale centralised approach suits non-renewable generation. In contrast, a small wind turbine or photovoltaic panel does not provide much energy on its own, but the total energy can scale to large quantities of generation when these are connected within a distributed framework in large numbers. Decentralised networks, being a middle ground between centralised and distributed networks, are therefore suited to smaller generators or larger renewable generation (e.g. a solar farm).

Managing resources used for energy generation presents its own challenges. Sustainability of a particular energy resource will depend on location, legal policies and economics of a region. Wood, for example, is used in biomass, which may become sustainable with well-planned schemes to replant uprooted trees. Uprooting trees without replanting them will lead to the complete depletion of that resource.

The use of fossil fuel energy in centralised systems is unsustainable in the long term as eventually the planet will be depleted of those resources. In contrast, renewable energy relies on sustainable energy sources (sunbeams, winds, geothermal stones, tides, etc). As well as sustainability, countries are beginning to manage their resources with respect to carbon emissions which are significantly lower in renewable energy generation. Overall, a shift towards distributed frameworks is expected, with non-renewable resources eventually going to be depleted and net-zero carbon energy goals forming. In the UK, legislation requires 100% carbon emission reduction relative to 1990 levels by 2050 [13].

7.2.3 Coordinated Structures from the lens of Security

When considering security features of the frameworks, it is worth noting that should a power fault occur in a centralised or decentralised network that all or some of the network may be negatively impacted by the generation loss. A real example is from 9^{th} August 2019. Initially triggered by a thunder strike, generation unit losses affected one million customers, health care buildings, transport and water facilities up to two days after the initial event [14]. The distributed framework is more forgiving when experiencing a fault. Each unit in the network is connected to multiple other units. A power unit can fail and rather than affecting the rest of the network it can be disconnected so the failure does not propagate through the network.

7.3 Proper Control Methods and Techniques for Active Buildings

In the previous sections, we have discussed the purpose of control and systems that might need controlling. In this section, the control methods and techniques that are applicable for active buildings are introduced.

7.3.1 Conventional Control and Monitoring Methods

Supervisory Control and Data Acquisition

Supervisory control and data acquisition (SCADA) is a control system architecture used in industry for monitoring and control of power systems, smart grids and also power generation and transmission. It is also used in building control and for other public infrastructures such as traffic lights and water management systems. SCADA uses wired and wireless communication across four network layers. At the highest level, an operator can communicate with the process layer via the *Human-machine Interface* (HMI). The process layer forwards this onto logic devices; the *Remote Terminal Units* (RTUs) and *programmable logic controllers* (PLCs) which can use the given information for aggregate control of field devices. At the lowest level *field devices* control and monitor the physical processes being observed by the system. These are sensors, pumps and other low-level pieces of equipment, that the observer may use to control the system as a whole. These devices provide feedback to SCADA via the HMI which helps check if the actual behaviour matches the desired behaviour [15], see Fig. 7.4.

As automation within industry increases and costs of operation reduces, the use of SCADA systems is expected to keep rising. However, the rise of the *Internet of Things* (IoT) has also impacted SCADA systems, and a transition from onsite and standalone systems to internet connected and remotely accessible systems is occurring. SCADA systems were never designed for network connectivity or network security. The focus was on reliability and device protection by isolating devices. SCADA is a complex system and with many inter-dependencies, so as devices go online, SCADA may become vulnerable. Another security concern, is that due to high installation costs,

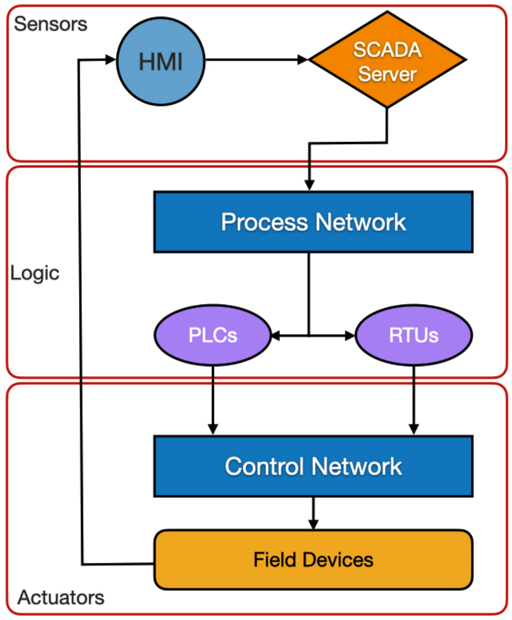

Fig. 7.4: Control system architecture of SCADA.

they remain operational for 8 – 15 years [16]. Relying on possible outdated or legacy systems could leave entry points to cyber-attacks.

PID Control

The most common industrial controller choice is the *PID controller*, an acronym for proportional (P), integral (I), derivative (D) control [17]. This is due to the simplicity of its operation and tuning, as well as its widespread use. PID controllers use the current and previous error measurements of a system for regulation. The system error is the difference between a reference value of the system and the equivalent measured value. Tuning of the system can be completed by adjusting the constant values, P, I and D, of the controller, which affect each of proportional, integral and derivative areas of the control equation respectively, as in Eq. 7.1.

In Fig. 7.5, an example plant, is shown with a PID controller. The system error, e, is fed into the PID controller. The PID controller uses e, the time step, t, and constants P, I, and D, to choose a new system input u, as shown in Eq. 7.1. The new input is passed to the system which produces the new output Y. The values Y_{ref} and Y_m represent the expected output of the system and the measured output respectively

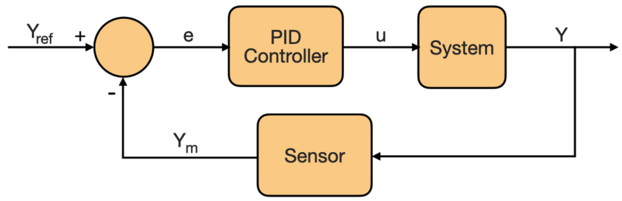

Fig. 7.5: PID controller - the system output is measured and checked against a reference value.

$$u(t+1) = Pe(t) + I\sum_{t=1}^{t} e(i) + D(e(t) - e(t-1)) \tag{7.1}$$

PID control is valuable for *Single-Input Single-Output (SISO)* systems and uncoupled two input two output systems. For *Multi-Input Multi-Output (MIMO)*, other techniques such as Model Predictive Control are more valuable [18, 19].

7.3.2 Model Predictive Control

Model Predictive Control (MPC), also called *Receding Horizon Control (RHC)* [20], is a control technique with some freedom involved in its implementation and is one of the fastest growing control techniques. MPC has a broad range of applications such as clinical anaesthesia, the cement industry, and robotics [21]. MPC algorithms use a model to represent a system, and attempt to minimise a cost function. Although, the implementation can be different, there are three important consistent ideas involved in MPC:

1. Use of a process model to create a prediction horizon;
2. Calculate a set of future control signals;
3. Use a receding strategy - the first value in the control sequence is applied to the process as it moves forward in time. At each time step a new prediction horizon and set of future control signals are calculated.

MPC uses a model to predict the future plant outputs. This is also known as the prediction horizon, $y(t+k \mid t)$ for $k = 1, \ldots, N$, where t is the current time step, and y is a set of future outputs for a time horizon, N. This is calculated using past control signals, u, past outputs, y, up until current time step, t, as well as predicted future control signals.

The set of future control signals is represented by $u(t+k \mid t)$ for $k = 0, \ldots, N$. From this, a control sequence which minimises the objective function at each time step, $t+k$, is also known. The objective function consists of a cost function and system constraints. It attempts to keep the process near to the reference value, $w(t+k)$. The optimisation uses the error between the reference values and the measured values.

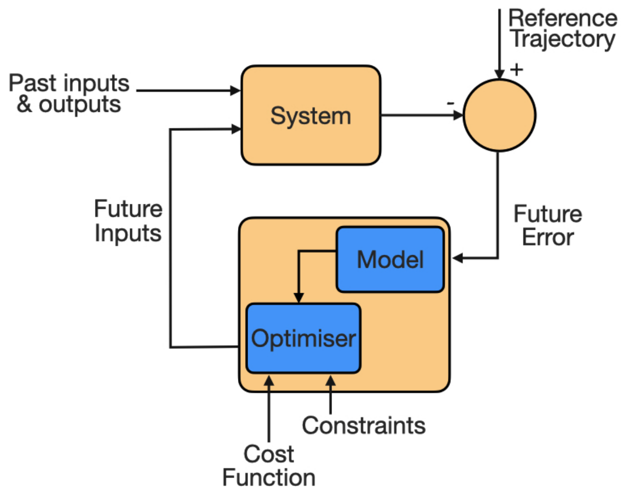

Fig. 7.6: Model Predictive Control - a model based control approach which predicts the next input that should be used. The model then fully updates for the next time step.

Calculating the optimal control signal considers the system constraints and the cost function. The overall cost function consists of the sequence of manipulated values (Z_k), a cost function for tracking error (J_y), a cost function for manipulated variable tracking (J_u), a cost function for change in manipulated variables ($J_{\delta u}$) and a cost function for constraint violations (J_ϵ) (see Eq. 7.2).

$$J(Z_k) = J_y(Z_k) + J_u(Z_k) + J_{\delta u}(Z_k) + J_\epsilon(Z_k) \tag{7.2}$$

The real-time solver computes the future sequence of manipulated variables, but only one control signal is exported for the next time step. At time t the control signal $u(t \mid t)$ is applied to the process. The remaining values of the sequence are discarded. For the next time step $t+1$, a new prediction horizon and sequence of control signals are calculated. Therefore, at time $t+1$ the control signal $u(t+1 \mid t+1)$ is used. This is more optimal than using the value $u(t+1 \mid t)$ computed in the first sequence since it considers the newly measured output value $y(t+1)$ which was unknown to the sequence at time step t. This is known as the receding strategy. Therefore as t increases the accuracy of MPC should improve as it has its predicted values corrected by real measurements.

MPC is advantageous as it is relatively simple to understand for those without control system experience and has applications in a wide variety of systems including unstable systems or those with complex dynamics. MPC deals easily with multi-variable processes, and can compensate for dead time. Using feed-forward control it is able to compensate for uncertainties in the system.

There are two major drawbacks. The first is that the required computation in the MPC algorithm can be very heavy. Implicit MPC is run online in the microprocessor, and there is no benefit in computing the best control signal after the time step it was needed has already passed. It is possible to compute the MPC algorithm offline, and store these values in lookup tables. For explicit MPC this is done and then the tables are imported to the microprocessor [18]. The second drawback, is the reliance on an accurate process model. MPC is computed *a priori*, with prior knowledge of the process. If the model used to compute the MPC algorithm has the wrong system dynamics, then the difference in the model and the real system will create discrepancies between the predicted control and the real control of the process. Although MPC is designed to adapt to errors in the model, the more significant the errors are the more difficult it will be for the MPC to correct them.

For those wishing to investigate MPC in more depth in the context of buildings, the review article [22] is helpful.

7.3.3 Multi-Agent System Approaches

An energy system like the one in an active building could be alternatively described as a group of components which interact to produce a reliable service at the lowest cost to consumers. This description involves decomposing an energy system into smaller structures which interact with one another. Multi-agent system (MAS) approaches support this framework for both modelling and control, and in [1] a definition for MAS is given:

> "A multi-agent system is a system composed of a collection of autonomous and interacting entities called agents, evolving in an environment where they can automatically perceive and act to satisfy their needs and objectives."

Agents receive information from the environment precepts through sensors. Such as a building with a PV panel on it's roof might have a sensor which detects the amount of solar irradiance currently being absorbed. This information is then passed through the agents logic which decides what action should be taken, in our example this might be whether to sell that energy to the grid if it is in excess of the buildings own current energy requirements. This choice is passed to the actuators which complete the task so the decided action occurs within the environment, potentially changing the global state of the environment. An example of this can be seen in Fig. 7.7. As the agents are separated from the environment they are in, MAS is a *distributed* approach to the energy management. MAS can also be described as *proactive*, the agents follow their own objectives or ones that cover the whole system, and *social*, the agents interact via cooperation or competition dependent on their objectives. For the MAS approach to be effective, an understanding of agent-agent and agent-environment interactions is paramount.

There are three main types of agents within the MAS framework. The first is the *reactive agent* which is the most basic and has minimal responses to environment precepts. They provide fast responses when needed and also have useful results

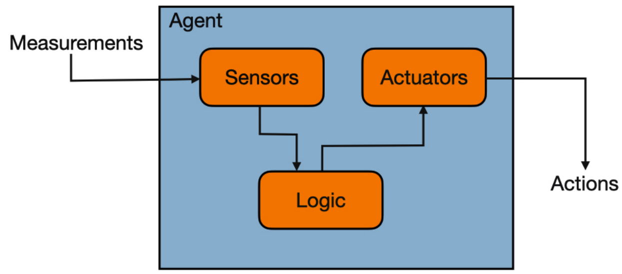

Fig. 7.7: A model of an agent which uses measurements of the environment to impact the agent's choice of actions.

when modelled as a large group of simple agents, rather than a single large agent. An example of a reactive agent is an EV which connects or disconnects to the grid without using any smart charging scheme. Modelling charging behaviour of a large group of EVs provides interesting results.

An *intelligent agent* has more functionality than a reactive agent since it is able to use its resources to achieve some objective. An example of an intelligent agent is a smart building with an energy management system (EMS), the agent uses its resources to satisfy the control objectives of its EMS.

The final agent is a *learning agent*, this agent gains information by analysing the outcome of their actions. The learning agent will have a good grasp of the environment it is in, which is used for decision making. The aforementioned smart building example, could become a learning agent by predicting the behaviour of those who use the building, and utilising this for its resource management.

MAS approach to control is bottom-up, due to the local knowledge of its agents and the flexible interactions which they have. The agents only know what they need to know, and this reduced data transfer across the network, making MAS scalable. It is unnecessary for two agents that will never interact to be aware of one another. The agents adapt to the situations they are in, making them flexible to faults where neighbouring agents fail or when new agents are added to the neighbourhood.

Other system approaches rely on predicting network changes and have costs associated with maintenance or redesign the network. MAS can avoid these costs. As these agents are cooperating or competing with one another autonomously the control approach required for the system is to distribute the control tasks between the agents. Decisions are made locally, with neighbouring agents grouping to form microgrids. The microgrids can then act as agents themselves to satisfy the global control objectives of the whole grid in a decentralised structure.

Difficulties of MAS revolve around the proactivity of the system. As each agent has a local objective and groups produce global objectives, it may occur that competing objectives arise. These challenges make distributed control more complex than the centralised control approach where an action would be demanded to suit the global specification. To help with this a clear communication framework between agents is needed, and definitions of the roles of the agents within the network to resolve objectives. This also benefits future hardware that will be incorporated into the networks.

7.3.4 Artificial Intelligence and Data Driven Control

Data driven control (DDC) approaches, including *machine learning* (ML) control are growing in prominence. Having been originally developed for static data, the methods and algorithms have been shown to be equally valuable when considering dynamic systems. The broad range of techniques allows ML control to optimise both linear and nonlinear systems. The most notable techniques that will be discussed in this section are: a) *reinforcement learning (RL)* and *artificial neural networks (ANN)* for designing control strategies; and b) *genetic algorithms (GA)* and *genetic programming (GP)* for finding a parameterised controller.

A major benefit of ML, compared to other control techniques already discussed in this chapter, is the possibility for model free control. Real-world control problems are especially difficult because they involve highly nonlinear dynamics, with an objective of maximising or minimising a certain property. System identification techniques may be impractical due to cost, complexity or other reasons (human based systems have ethical considerations). ML relies on sensor data only to optimise an objective function. It is a powerful tool when system models are unavailable. Some example real-world fields that ML techniques can help include epidemiology, robotics and fluid dynamics, but there are many more [23].

For the system that will be controlled, a cost function is minimised to satisfy the system objective. The ML controller will pass inputs to the physical system which reduce this cost value. To do this, a best strategy must be learned by the ML controller. This is completed offline, with the outputs of the system and the cost function passing to the ML controller. Using this data and differing ML algorithms a control strategy is formed. With more data the controller can gain more experience and provides better control functions.

Reinforcement Learning - Experience Based Control

RL is an algorithm that acquires experience over time to improve its control policy. *Markov decision processes (MDPs)* are the most commonly used framework for RL. MDPs incorporate uncertainty in their description of system dynamics and control laws. This promotes optimisation and exploration of the state space in equal measure.

RL is run by an agent who is in charge of choosing the control policy. Typically, the solution to an RL control strategy is binary, either the strategy is successful or it is not. To improve this, algorithms have formed a *value function*, these are known as Q-learning algorithms. This function denotes the value of success represented by the current state, and can be considered proportional to the likelihood of a winning

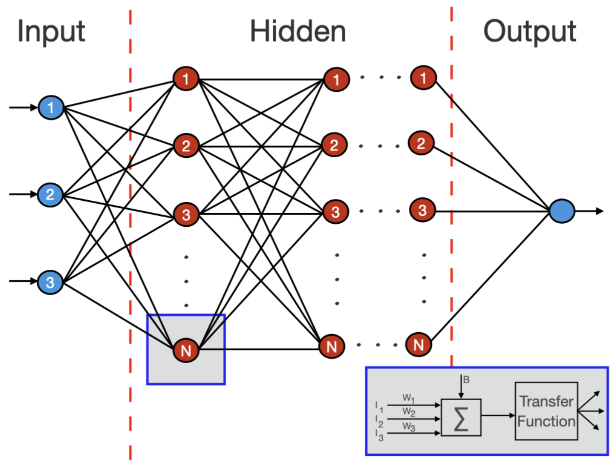

Fig. 7.8: Artificial Neural Network with multiple hidden layers. A model representation of Eq. 7.3 is also given with inputs, I, bias, B, and weights, w.

strategy. For example, if a good control policy is in effect, the value function for the overall system state should be high. As RL learns from experience, the algorithm might initially choose low scoring strategies, but after many iterations will learn to choose higher scoring control strategies with more chance of ultimately being successful [24].

Artificial Neural Networks - Data Driven Control

ANN were developed from the *perceptron*, a mathematical model of a synapse in the brain. Perceptrons are quite limited, only returning a binary value, but when layered on top of one another they can learn system behaviours. ANN are easy to create and implement. As more research into ANN is done, other techniques such as *recurrent neural networks* and *deep neural networks* have been developed. In essence, ANN is used to predict an output for any given input based on all previously known inputs and outputs.

Neural networks consist of three sections, see Fig. 7.8. The first section is the input layer, where the system inputs are passed into the algorithm. The inputs are then passed to the hidden layer, with N-layers of neurons. The neurons in the first layer receive all the normalised inputs from the input layer and compute an output value which is then passed to each neuron in the next layer. Those neurons complete the same process, with the inputs they receive, until all layers have computed some output. The final output values are then passed to the output layer for the overall output prediction. From previous prediction errors, the ANN can adjust weights associated with each layer to improve future performance, this is known as *backpropogation* [25].

$$y = tansig(B + \sum_{n=1}^{n}(I_n \times w_n)) \tag{7.3}$$

At an individual neuron the output formula is given in Eq. 7.3. y is the neurons output, B is a bias value, I_n is the input and w_n is the weight associated to the respective input. The summation of inputs multiplied by their weights is calculated, then a bias value is added. This result is then passed through a transfer function such as $tansig()$, $logsig()$ or another, depending on the specific algorithm in use. In practical terms, there are many useful tools already available to help with the creation of an ANN. One example is the neural network toolbox for MATLAB [26].

To run the ANN algorithm, a large data set is required of measured input and output values. The data is divided into a training data set (majority of the data) and a validation data set (remaining data). The strength of ANN lies in its training algorithm, which decides the weights and the bias to apply to each neuron. The algorithm optimises the values of w and b for the data, using a mean square error or sum of errors in the process. Once the training has finished forming the model, the algorithm can predict the expected output values of a given input.

Occasionally, the ANN model focuses on smaller details present in the training set over the general trends of the input to output mapping. This is known as *overfitting* and to avoid this, validation data is used. The model is run using the inputs from the validation data, but the output values remain hidden. The model predicts outputs from the validation data that can be checked against the actual outputs from that data which had been hidden. If the there is a significant error in these predictions then it is likely that overfitting has occurred within the model, and the training should be redone.

Parameter Synthesis Using the Genetic Algorithm

When the structure of a control law is given but the parameters are unknown, ML control becomes a parameter identification. Genetic algorithms (GAs) are meta-heuristic optimisation approaches based on the laws of natural selection from evolutionary biology, and also have applications outside of ML control. With each new generation, the most successful (or fittest) individuals climb to the top of the rankings. An individual is a member of a generation, k, with a random assignment of parameter values. This combination of parameters is called its DNA, and are written as a numeric sequence. These parameters are what the algorithm tries to optimise via a set of *genetic rules*.

The fittest individuals in a generation have minimised their cost function or error scores. Once a generation has been computed, all the individuals are ordered by cost function value. A probability is assigned to each individual, relative to this cost value. A lower cost will equate to a higher probability of being selected for the next generation, $k + 1$. Optionally, the top x number of individuals can be immediately moved to $k + 1$, with probability 1. This is known as *elitism* and for the remaining individuals, any who are selected for generation $k + 1$ will undergo one of three genetic rules:

1. *Replication* - The individual is moved to generation $k + 1$ immediately with no modifications.
2. *Crossover* - Two individuals swap a portion of their DNA, then both move to generation $k + 1$ in their new modified forms.
3. *Mutation* - An individual has its DNA randomly modified before moving to generation $k + 1$.

The remaining individuals are then discarded. As the GA iterates through the generations, the fittest individuals with lowest cost function scores appear. GA stops when these individuals have converged or a stopping condition is met.

GA is useful as it does not require the iterations of a brute force algorithm, trying every possible set of parameters, and it scales to high dimension spaces better than other algorithms, such as Monte Carlo sampling. However, GA does not guarantee converging on an optimal solution. Additionally, choosing the size of the population and the number of generations affect the algorithms performance.

Genetic Programming - Control Law Identification

GP is an extension of GA, but can be treated as its own control technique. It is used to optimise the parameters of the system and even the structure of the system. GP can also find appropriate control laws. It does this by completing a GA approach for different structures as well as different parameters for those structures.

GP uses a recursive tree structure to encode the complex functions of sensor signals. By using such a generalised framework it is possible to identify the structures of nonlinear control laws for highly nonlinear systems. GP works especially well on problems where the solutions can be checked quickly, this allows GP to test a large number of individual control laws for their suitability. A successful GP approach will find not just the optimal parameters for the model but also the optimal model to use.

7.3.5 Game Theoretic Approaches

Game theory is an optimisation method with mathematical foundations. Entities, or players, attempt to achieve individual objectives and take actions to complete them. Deciding on the best actions to take, and the outcome of those actions, is the essence of game theory. A player’s utility is the value of their success in the game, relative to the other players. Depending on the strategy, a players utility can increase or decrease. Game theoretic approaches usual try and manage the utility of the different players to satisfy an overall global objective. There are two main areas of game theory; noncooperative and cooperative approaches.

Under these two main umbrellas, games can also be either dynamic or static. Dynamic games consider time when playing, and thus allow for consecutive player moves. Static games on the other hand do not consider time, and so player moves are either simultaneous or alternating.

In *noncooperative game theory*, players cannot communicate with one another and so must choose actions without coordinating their choices [27]. The global solution to the game often takes the form of a *Nash equilibrium*. A Nash equilibrium is a state where no player can improve their utility. Essentially, this is a draw between all players, and changing the strategy will only result in worsened performance for the player who changes strategy. Finding such an equilibrium is not always simple and certainly not guaranteed [28].

Cooperative game theory allows communications between the players. Since the players can communicate, they have the option of whether they wish to cooperate with one another or not. This leads to two main cooperative strategies. Firstly, the *Nash bargaining strategy* where the players communicate with one another to determine a contract under which they agree to cooperate. This strategy allows for competition, but with some agreed trading. Secondly, the *coalition strategy* where the players group together as one coalition. This strategy is fully cooperative, and the players unite under the same objective. Once it is clear which type of game is being played by the players, the in-game strategies can be considered [29].

There are three essential parts to a players turn, no matter which game they are under. The player must consider the global game state to understand the current utility and set of actions that player has. The player must estimate their prospective utility for their actions. The player updates their strategy based on those observations and chooses an action. There is some variety regarding the algorithms which are used to complete these three steps:

- *Best response dynamics* - simplest approach, chooses the action which maximises the players utility but does not guarantee convergence to an equilibrium point.
- *Fictitious play* - considers the actions of all players before choosing an action. For zero-sum games, it will always converge to a Nash equilibrium.
- *Regret matching* - a strategy which chooses the least detrimental action, as opposed to the most beneficial action.

There are also other algorithms such as reinforcement learning and stochastic learning that can be chosen [30].

An assumption from classical game theory is that the players are rational, in reality this is not necessarily the case. Small changes from the optimal strategy via non-optimal play could have disastrous knock-on consequences. To combat this there are analytical techniques to avoid such faults from occurring. Overall, the robustness of the algorithm design and model is essential to safe operations.

Game theory approaches have many power system applications. Some examples include; cooperative energy exchange, distributed control of microgrids and smart grid communication technologies. Classical approaches of game theory for demand-side management focus on the relationship between individual buildings and the

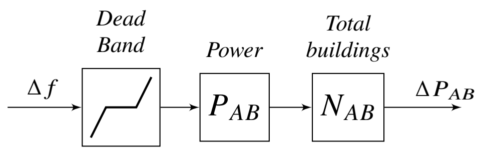

Fig. 7.9: A simple model for participation of active buildings in frequency response services.

operator and the optimal strategy for buildings is to use an aggregate behaviour when negotiating with the operator [31].

7.4 Coordinated Control of Active Buildings: Case Studies

With the development of increased renewable technologies, we are seeing a transition from centralised coordination and control strategies to distributed ones. Power networks have increased complexity and decreased predictability, proportional to the increased distribution. For control systems to remain smart, the system intelligence must come from the information sharing and cooperation of the components inside the network [1]. This is appropriate when we consider active buildings as they are able to take part in both the generation and demand as prosumers.

In this section, case studies of active buildings within the three different coordination frameworks are provided.

7.4.1 Centralised

For this case study, we will analyse the frequency response from the *load-frequency control* (LFC) model as well as the response from active buildings included in the network that can be used for frequency response. As this is a centralised network, we consider a single controller which is able to cut the power to buildings when necessary in order to respond to load disturbances. Fig. 7.9 shows a simple model of the active building behaviour; when the frequency leaves the deadband interval the controller cuts the power to buildings where necessary to reduce their load on the system and provide frequency response services.

In this scenario, the buildings are only considered to be loads on the network. Due to privacy concerns, the controller does not have permission to regulate the smart devices within an active building unless the building is part of a scheme such as aggregation. Aggregation is not a centralised control scheme and therefore the buildings can only contribute to frequency response by being removed from the network entirely and tasked with relying on their own energy generation.

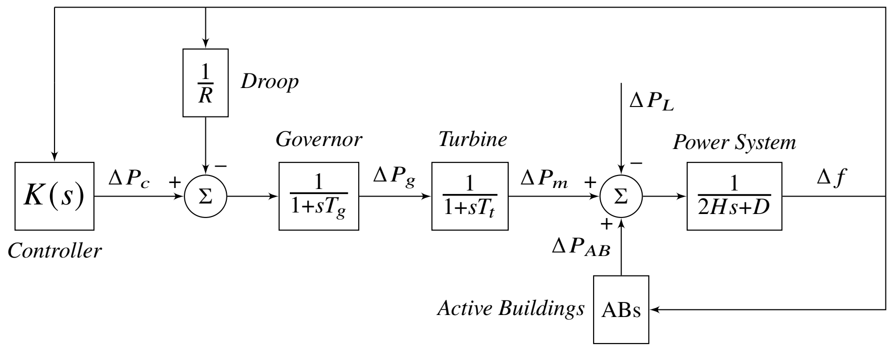

Fig. 7.10: A simple LFC model, with generator dynamics taken from [32] with added active buildings for centralised control.

Finding an optimal network controller can reduce the likelihood of frequency disturbances causing the load shedding of active buildings. In this case study, a PID controller will be used to provide an improved frequency response on the simple controller given in [32]. The system values used for the simulation can be seen in Table 7.1. The step load disturbance for the simulations is 0.02 pu.

Table 7.1: Values used for simulation adapted from [32].

Symbol	D	$2H$	R	T_g	T_t	P_{AB}	N_{AB}	*deadband*
Value	0.015	0.1667	3.00	0.08	0.40	2000	25	0.0 ± 0.01
Units	*pu/Hz*	*pu s*	*Hz/pu*	*pu s*	*pu s*	*kW*	-	*Hz*

Power systems are highly nonlinear and uncertain, and regulation combines fast voltage and rotor angle responses with slower frequency responses. By only considering frequency response, a simpler linear model under a load disturbance can be used for frequency control synthesis. The relationship of the generator-load dynamic can be described as:

$$\Delta P_m(t) - \Delta P_L(t) = 2H\frac{\delta}{\delta t}\Delta f(t) + D\Delta f(t) \tag{7.4}$$

Δf is the frequency deviation, ΔP_m is the mechanical power changes, ΔP_L is the load changes, H is the inertia constant and D is the load damp coefficient, where 1% change in frequency is equivalent to D% change in load. By applying the Laplace transform on Eq. 7.4, a new equation is formed.

$$\Delta P_m(s) - \Delta P_L(s) = 2Hs\Delta f(s) + D\Delta f(s) \tag{7.5}$$

Eq. 7.5 can be used in Simulink to produce a graphical block diagram of the system, Fig. 7.10. The block diagram, taken from [32], represents the LFC synthesis

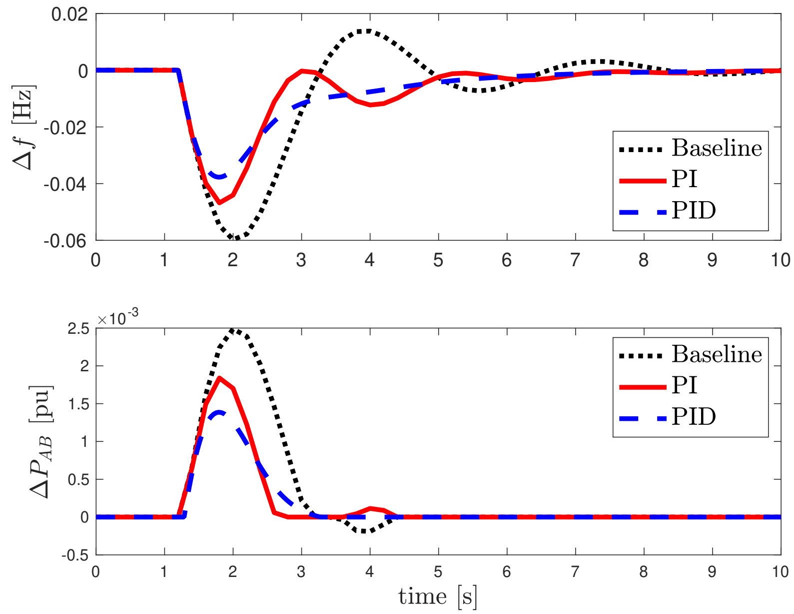

Fig. 7.11: Frequency response using PID, PI and baseline controller from [32] **(top)**. Active building load change **(bottom)**.

of a non-reheat steam generator unit and includes governor, generator, turbine, load and secondary control characteristics. The slow boiler dynamics, and fast generator dynamics are ignored. A speed droop characteristic, R, is included which shows the speed regulation due to governor actions. K is the system controller.

For the baseline case the controller $K(s)$ is $-0.3/s$ and for PID control the values are set to $P = -0.3$, $I = -0.3$ and $D = -0.1$. Another PID controller omitting the derivative, known as a *proportional-integral* (PI) *controller*, is also provided in Fig. 7.11. It is clear to see from the figure that the PID controller improves the frequency response of the system, having a lower minimum frequency value and also reducing the need to use the active building power as part of the control. This leads to less disconnections of buildings from the network which is beneficial to both building users and the network. At the minimum frequency measurement, the baseline controller requires the equivalent P_{AB} power of 25 ABs, while for the PID controller this is nearly halved. Even the simpler PI controller provides a large improvement on the baseline controller taken from [32]. The PID controller is simple to implement and for SISO systems such as the basic frequency response model it is a valuable control technique.

Realistically, after a frequency disturbance of only -0.06 Hz to have a *low frequency demand disconnection* (LFDD) scheme trigger would mean the network was

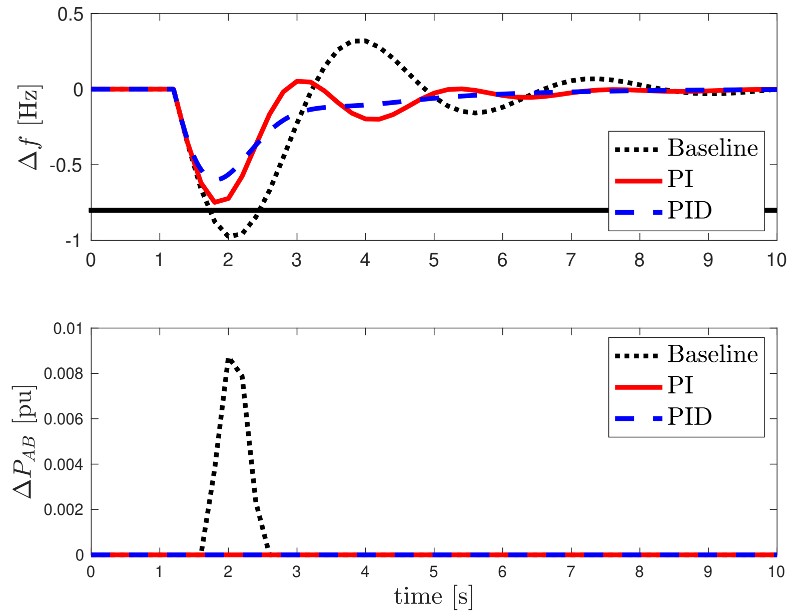

Fig. 7.12: Frequency response using PID, PI and baseline controller, the horizontal bar marks where the LFDD scheme triggers **(top)**. Active building load change **(bottom)**.

very unstable. From literature, in the UK it is at -0.8 Hz where the LFDD schemes are imposed [33]. We will adjust the values of the simulation so that the deadband is 0 ± 0.8 Hz and the load disturbance is 0.3 pu. Now the active buildings will only be disconnected from the network when the frequency falls to -0.8 Hz or below. In Fig. 7.12, it can be clearly seen that for the baseline controller the LFDD scheme triggers at -0.8 Hz, while for both the PI and PID controllers the network protects the active buildings from needing to be disconnected. In this case study, it can be seen that control and management of ABs from the side of the power grid can reduce the need for LFDD schemes. Additionally, it can be implied that ABs which manage their consumption from the power grid will become less of a burden as they have reduced the load needing to be controlled.

7.4.2 Decentralised

In the second case study, we will consider a system with a decentralised structure and analyse the role of active buildings in frequency control of such a system. In the

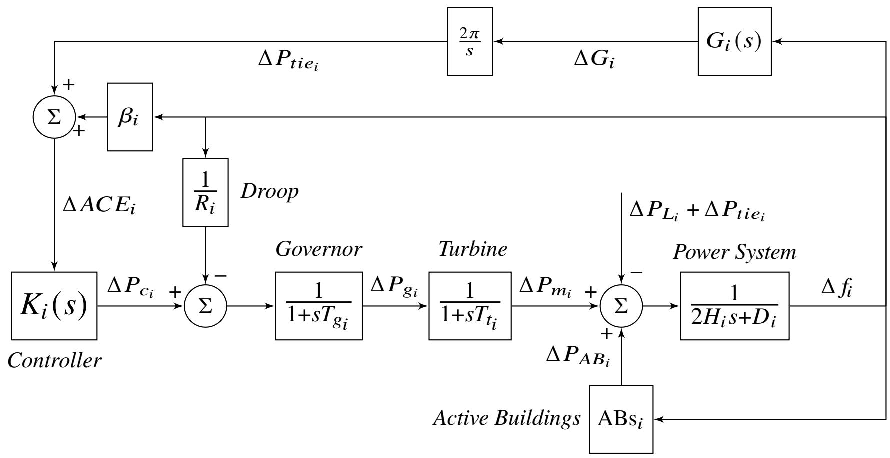

Fig. 7.13: A simple LFC model, with generator dynamics taken from [32] with added active buildings for centralised control. The diagram shows area i, one of the areas inside the 3-area system.

decentralised structure, the system is divided into smaller areas, which each have their own control method of the frequency. In this way, a global control is completed by the interactions of the smaller control areas. For this case study, we will consider 3 areas which each use MPC as their control technique.

In the centralised structure, it was not possible for the active buildings to respond to the frequency events since the central controller did not have the permissions to adjust the consumption of the buildings. In decentralised control, buildings can sign up to the schemes run by aggregators. The aggregator is able to negotiate in the energy market on behalf of a significant number of buildings, thus reducing the energy costs of the buildings but also providing services to the power network. Having opted into the scheme, the buildings have given aggregators permissions to adjust their consumption values and can be controlled for demand-side response services. In this case study we consider that each area has 50 large buildings available for control with 1000 kW of controllable load.

A basic 3-area system for frequency response is provided in [32]. Fig. 7.13 shows one area indexed by i of the system. As well as maintaining the frequency of the area, a controller is needed to deal with the interchange of power between areas. This is done by adding the tie line flow deviations to the secondary control of the frequency area, ΔP_{tie_i}. The value of ΔP_{tie_i} is also included in the summation before the Power System block inside the diagram. For space, this value is combined with P_{L_i} in the diagram. $G(s)$ is used to calculate the amount of power given or received from the other areas, its formula is given in Eq. 7.6, where N is the number of areas, T_{ij} is the synchronising torque coefficient between areas i and j, and Δf is the frequency of area i or j respectively. The values used in the following simulations are shown in Table 7.2. In the simulations $T_{12} = T_{21} = 0.2$, $T_{13} = T_{31} = 0.25$, $T_{23} = T_{32} = 0.12$,

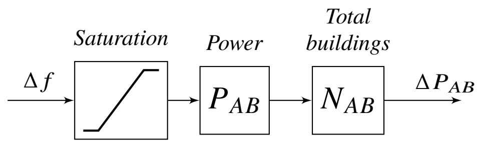

Fig. 7.14: The base controller for active building frequency response for the three area system.

$K_1(s) = -0.3/s$, $K_2(s) = -0.2/s$ and $K_3(s) = -0.4/s$.

$$\Delta G_i = \sum_{j=1, j\neq i}^{N} T_{ij}\Delta f_i - \sum_{j=1, j\neq i}^{N} T_{ij}\Delta f_j \tag{7.6}$$

Table 7.2: Values used for simulation adapted from [32].

Area	D_i	$2H_i$	R_i	T_{g_i}	T_{t_i}	β_i	ΔP_{L_i}
$i = 1$	0.015	0.1667	3.00	0.08	0.40	0.3483	0.3
$i = 2$	0.016	0.2017	2.73	0.06	0.44	0.3827	0.0
$i = 3$	0.015	0.1247	2.82	0.07	0.3	0.3692	0.3

Now that we defined the decentralised frequency response system that will be used, we can discuss the control technique. We will compare the performances when an aggregator uses a robust MPC controller and when it uses a simple controller based on a saturation of the frequency, see Fig. 7.14. The simple saturation controller, in future it will be referred to as the base controller, chooses an input u to apply to the system. u is the participation value of the buildings in the control, and in each area $P_{AB_i} = 1000$ kW and $N_{AB_i} = 50$. The saturation sets $u = -2 \times \Delta f$ Hz, but also if $\Delta f < -0.5$ Hz then $u = 1$ and if $\Delta f > -0.015$ Hz then $u = 0$. A buffer zone of 0.015 Hz is given to stop unnecessary fluctuations around the nominal value.

For this case study, we are considering a system experiencing a disturbance and desire to return the system back to the nominal value as quickly and smoothly as possible. An MPC control algorithm which suits this scenario is the robust optimisation framework of the MPC algorithm. Using YALMIP [34] and MATLAB, we will derive a controller for each area, that is robust enough for certain levels of disturbance.

The first step is to discretise the system. It is fairly simple to convert the transfer function model shown in Fig. 7.13 to a state space model. Consider each area has 5 states, f_i,P_{c_i},P_{g_i},P_{m_i}, and P_{tie_i}, each area has one input, the percentage participation of active buildings, and a disturbance, P_{L_i}. This gives a system with 15 states, 3 inputs and 3 disturbances that can be described in the form:

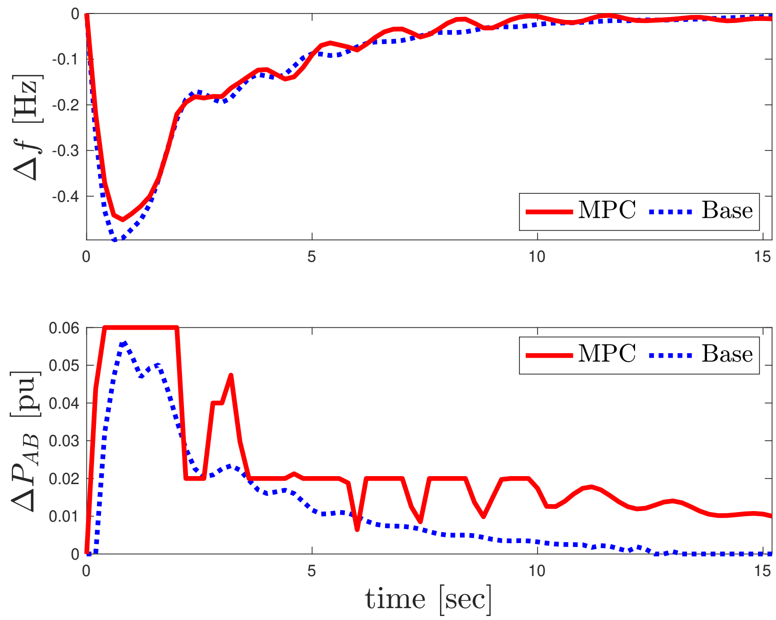

Fig. 7.15: Global frequency measurements **(top)** and participation measurements **(bottom)** for areas with $L1 = 0.3$ and $L3 = 0.3$.

$$\dot{x}(t) = Ax(t) + Bu(t) + B_w w(t)$$

where $x(t)$ is the state, $u(t)$ is the input and $w(t)$ is the system disturbance.

To discretise the system, and transform the continuous differential equations to discrete difference equations, a clever transformation is computed.

$$e^{\begin{bmatrix} A & B \\ 0 & 0 \end{bmatrix} T} = \begin{bmatrix} A_d & B_d \\ 0 & I \end{bmatrix}$$

where T is the sample step, A_d and B_d are the discretised state space matrices of A and B respectively, and I is the identity matrix. By treating the disturbance as an input, B_w can also be discretised using the same equation.

As previously described, the MPC algorithm has three steps.

1. Create a prediction horizon.
2. Calculate a set of future control signals.
3. Use a receding strategy to apply the current optimal control signal, then compute new control signals starting from the next time step.

Firstly, we create a prediction horizon, we set the program to look 10 time steps ahead when choosing the best input value for the system. We also set constraints

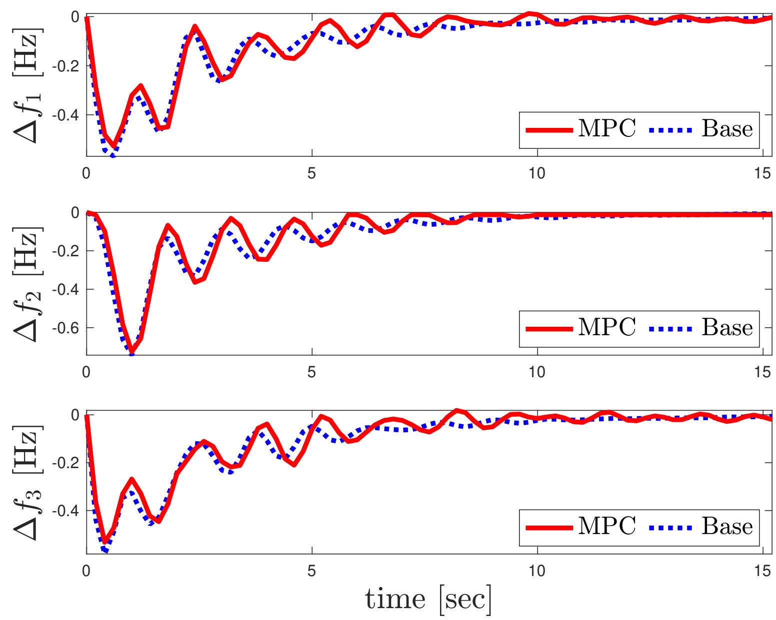

Fig. 7.16: Frequency measurements for areas 1 **(top)**, 2 **(middle)**, and 3 **(bottom)** with $L1 = 0.3$ and $L3 = 0.3$.

on the states and inputs, for this system the input is the participation value so it should only be between $0 \leq U \leq 1$. and the frequency in each area should not fall below the containment zone $-0.8 \leq Y_i$. For robust MPC we consider the disturbance and try and maintain optimal performance of the system considering the worst case scenario. By modelling the power loss in each area in the matrix B_w, we can set the disturbance W as $0 \leq W \leq 1$. The model will consider the worst case scenario when computing the optimal input values to choose, which is where $W = 1$.

Secondly, at each time step the algorithm computes a set of future control signals. This is done by considering the constraints put on the system and the control objective. If the constraints cannot be satisfied by any of the possible inputs then a non-feasible solution is found. For all the feasible solutions, an objective function is used to calculate the most optimal input of the group. The objective function adds weights to different control objectives. For this case study we wish to minimise Y, the frequencies of each area, and also minimise U, the input for each area. Minimising the frequency is prioritised over minimising the input so a weight of 0.01 is applied to the objective function for U. The objective O is shown in Eq. 7.7, where N is the number of areas.

$$O = \sum_{i=1}^{N} (\|Y_i\| + 0.01\|U_i\|). \tag{7.7}$$

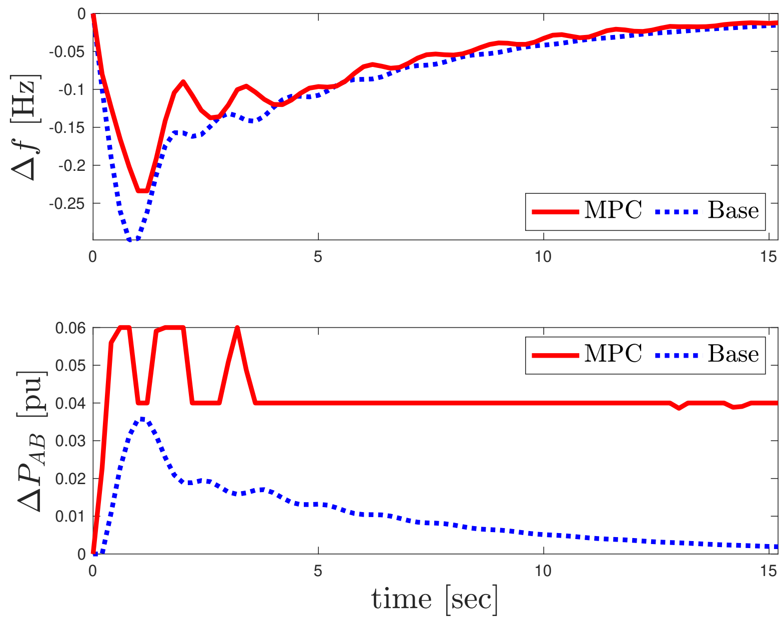

Fig. 7.17: Global frequency measurements **(top)** and participation measurements **(bottom)** for areas with $L2 = 0.3$.

Thirdly, By computing the objective function at the current time step, t_k, for the full prediction horizon, an optimal input value is found. This value is passed into the plant and the system evolves. The MPC algorithm now discards the previous computations and begins again this time with the current time step t_{k+1}.

In Fig. 7.15, it can be seen that the MPC algorithm provides a significant benefit for frequency response compared to the base controller. The disturbance on the system has a smaller impact than felt by the base controller. In Fig. 7.16, it can be seen that the MPC controller provides small improvements on the frequency response of all three areas of the system. Each area also satisfies the requirement of remaining outside of the containment zone. The MPC controller is robust with respect to the worst case scenario, finding an optimal solution in every time step.

This MPC controller is robust to a wide variety of disturbance scenarios. We will compute the same controller but change some of the system values - $L_1 = 0$, $L_2 = 0.3$ and $L_3 = 0$. In Fig. 7.17, a significant improvement over the base controller can be seen in this scenario. The overall contribution of the active buildings is also higher. Each area also has significant frequency response improvements over the base scenario, as seen in Fig. 7.18. It is also noteworthy, that in this scenario, area 2 has frequency transients similar to the other areas, yet for Fig. 7.16 there was a significant difference in the frequency transient of area 2 and both of the other areas.

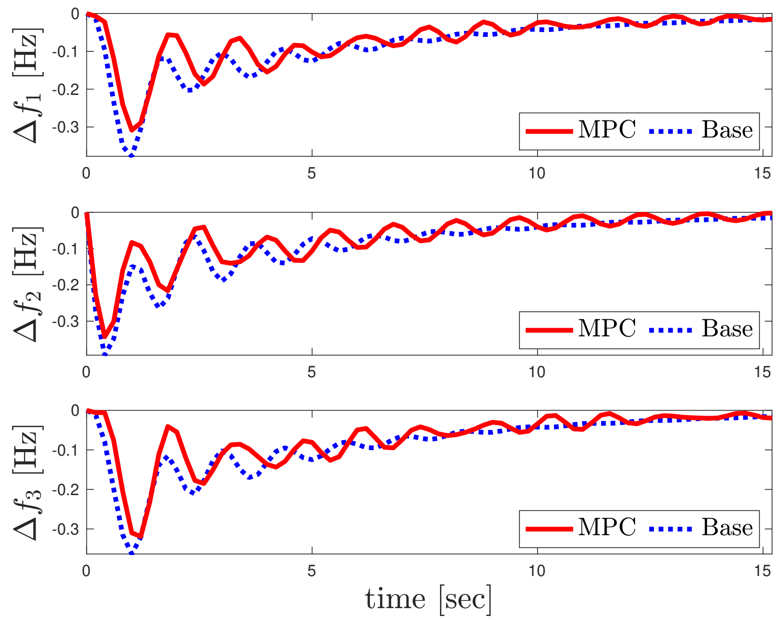

Fig. 7.18: Frequency measurements for areas 1 **(top)**, 2 **(middle)**, and 3 **(bottom)**, with $L_2 = 0.3$.

Overall, MPC is a viable and popular control technique which has benefits when controlling systems with disturbances and uncertainty. The use of a cost function in the algorithm means MPC can provide accurate control for MIMO systems, and optimise the system despite the additional complexities, relative to the centralised scenario. This technique is beneficial for controlling ABs as seen in the case study because the algorithm was able to control the transient frequency of the smart grid to minimise the system loss while also considering AB requirements attempting to minimise their contribution.

7.4.3 Distributed

For the case study of the distributed structure, we will use the MAS control technique. We will consider a population of homogeneous residential ABs, smaller than the large ABs used in the previous two case studies, which can act as reactive or intelligent agents in the system. The buildings are connected to the smart grid and each have an ESS and PV panel. It is assumed that active buildings store energy in their ESS for potential demand-side response services which have been generated from their PV panel or stored from the grid in a previous iteration of time. Each building knows the percentage charge of the ESS it is paired too, but this is not known by the smart grid globally. Instead, the smart grid broadcasts a signal containing the system frequency and the agents decide whether or not to respond.

Algorithm 1: Active Building Agent Logic

```
running = true;
while running do
    if f < -0.1 & c > 40 then
        inject power
    else
        do nothing
    end
end
```

The logic of each agent, as shown in Alg. 1, is setup to be quite simple. At each time step, the agent will check the global frequency value and compute whether or not to inject power into the network. If the global frequency is below -0.1 Hz and the building has at least 40% charge in its ESS then demand-side response services are provided. By discharging the battery to the power grid the battery will have

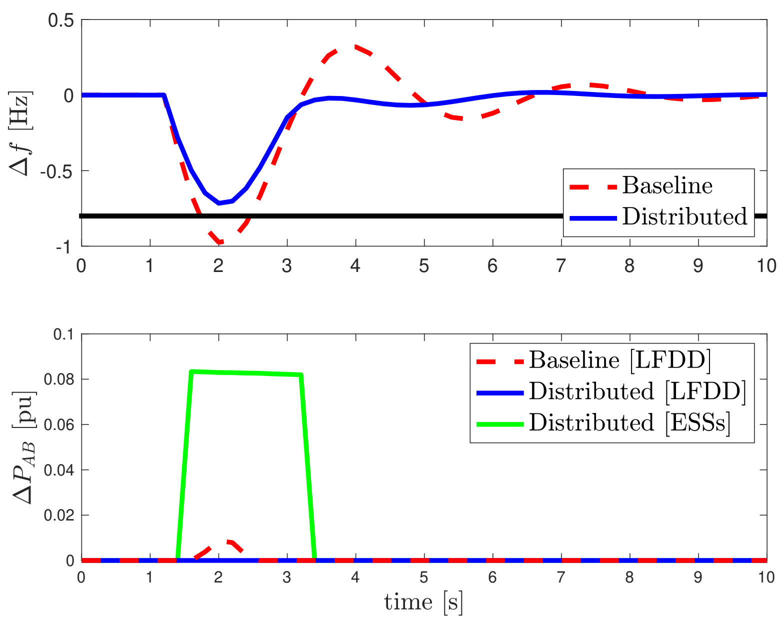

Fig. 7.19: Frequency response using distributed agent control and baseline control, the horizontal bar marks where the LFDD scheme triggers **(top)**. Power injection to system from active building agents **(bottom)**.

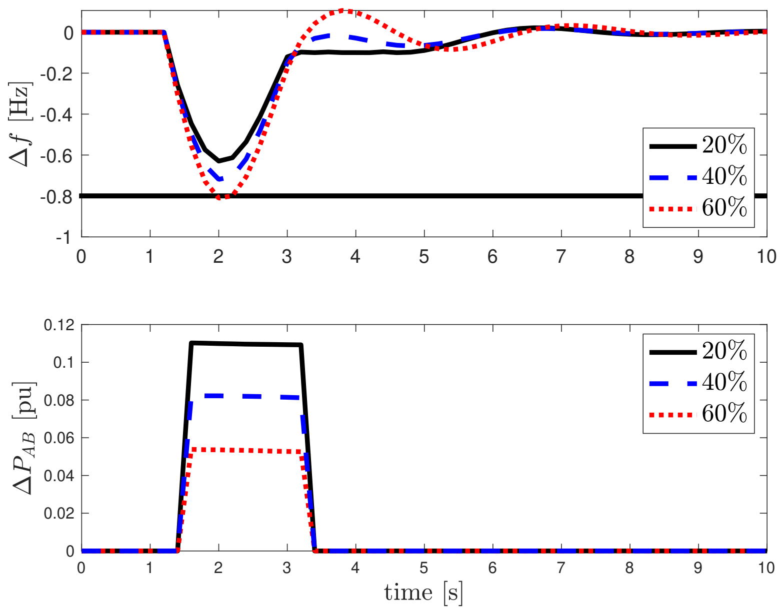

Fig. 7.20: Frequency response using distributed agent control, the horizontal bar marks where the LFDD scheme triggers **(top)**. Power injection to system from active building agents **(bottom)**.

less capacity remaining. Therefore, it is assumed that, at each time step, 1% of the available capacity is lost when discharging.

First, we will consider these active building agent response within the power system model described for the centralised controller. Taking the model from Fig. 7.10, and the values that produced Fig. 7.12, the model we consider is a single area model of the frequency which disconnects active buildings when the frequency drops below -0.8 Hz. We now consider that all the buildings in that network have an ESS, and that these buildings can inject their power into the power system if they decide to do so following the logic from Alg. 1. The percentage charge of each building's ESS is chosen to be a uniformly distributed random value between 0% and 100%. The power injection per second of each homogeneous ESS is set to 28 kW. In this scenario, we will consider a population of residential buildings, $P_{AB} = 10$ kW and $N = 5000$, to show that many AB agents can collaborate for demand-side response services.

From Fig. 7.19, it is clear that having addition demand-side response from multi-agent systems can greatly enhance the frequency response of the power network and avoid LFDD schemes. Additionally, in Fig. 7.20 it is seen how different MAS logics can affect the control. In the original scenario a charge of at least 40% was required for MAS demand-side response, c value in Alg. 1. The frequency response can be enhanced further when the charge is set to 20% or worsened when set to 60%. In this case study, very simple agent objectives were presented; the agent wishes to contribute to demand-side response to avoid the need for LFDD. However, the agent objectives can be complex and competing objectives may arise. To solve this issue, communication algorithms are needed for the agents.

Overall, the distributed approach, such as the use of MAS, is viable for future applications because it is highly scalable. It is expected that the AB penetration in the power network will increase and so independent agents which can respond to the state of the global power network are valuable. This reduces the need for a complex centralised or decentralised control strategy that can rely on accurate models of the power network.

7.5 Status Quo, Challenges, and Outlook

In this chapter, we have looked at the control and management of active buildings. In particular, we have discussed the coordination structures and frameworks which consist of a system structure; centralised, decentralised or distributed, and an energy resource type; renewable or non-renewable. We discussed various control methods that can be used for active building systems, the most popular being PID control and MPC, but other viable techniques are also emerging such as MAS control and DDC. Finally, we provided examples for the control and management of the active buildings at each of the centralised, decentralised and distributed system frameworks. Two future research directions are as follows:

- Firstly, the smart grid is evolving to be more distributed and electrified as more passive buildings become ABs. These new ABs and other grid components will join the power network. This increased load will add a strain to the network unless control schemes are developed to safely manage ABs while also keeping AB users satisfied. The increase in system loads leads to a larger complex network and the requirement of having scalable methods for grid control. A scalable strategy discussed in this chapter was MAS, but further research into the management of ABs and communication between competing and cooperative AB objectives needs further research.
- Secondly, as the smart grid becomes more electrified, and with the increase in ABs, it will become more complex. This complexity will create further nonlinearity within the system and be more difficult to model. Many of the current control techniques are model based, such as MPC, and so rely on accurate underlying models of a system. As complexity increases a shift towards model-free or data-driven modelling as a prospective option has increased. These methods rely on large quantities of data to represent the system which can be difficult to acquire as well as potentially losing some system understanding provided by previous model's physical descriptions. Further research into efficient data-driven control

schemes possibly combined or guided by simplified models are needed to manage ABs in a smart grid.

References

1. R. Roche, F. Lauri, B. Blunier, A. Miraoui, and A. Koukam, *Multi-Agent Technology for Power System Control*. Springer London, 2013.
2. M. M. Rathore, A. Ahmad, A. Paul, and S. Rho, “Urban planning and building smart cities based on the Internet of Things using Big Data analytics,” *Computer Networks*, 2016.
3. J. Reynolds, “Real-time and semantic energy management across buildings in a district configuration,” Ph.D. dissertation, Cardiff University, 2019.
4. L. Miao, G. Wei, X. Fang, and J. Risheng, “The strategy of the voltage control in smart grid based on modern control method and FPGA,” in *2015 34th Chinese Control Conference (CCC)*, 2015.
5. L. Ljung, “Perspectives on System Identification,” *Annual Reviews in Control*, vol. 34, 2010.
6. M. Ławryńczuk, “Nonlinear predictive control of a boiler-turbine unit: A state-space approach with successive on-line model linearisation and quadratic optimisation,” *ISA Transactions*, vol. 67, 2017.
7. H. Zhang, J. Zhou, Q. Sun, J. M. Guerrero, and D. Ma, “Data-Driven Control for Interlinked AC/DC Microgrids Via Model-Free Adaptive Control and Dual-Droop Control,” *IEEE Transactions on Smart Grid*, 2017.
8. L. Arnaboldi, R. M. Czekster, C. Morisset, and R. Metere, “Modelling Load-Changing Attacks in Cyber-Physical Systems,” *Electronic Notes in Theoretical Computer Science*, 2020.
9. J. Hu, G. Yang, K. Kok, Y. Xue, and H. W. Bindner, “Transactive control: a framework for operating power systems characterized by high penetration of distributed energy resources,” *Journal of Modern Power Systems and Clean Energy*, 2017.
10. A. M. Annaswamy, A. R. Malekpour, and S. Baros, “Emerging research topics in control for smart infrastructures,” *Annual Reviews in Control*, vol. 42, pp. 259–270, 2016.
11. C. Vezzoli, F. Ceschin, L. Osanjo, M. M’Rithaa, R. Moalosi, V. Nakazibwe, and J. C. Diehl, “Designing Sustainable Energy for All. Sustainable Product-Service System Design Applied to Distributed Renewable Energy,” *Designing Sustainable Energy for All*, 2018.
12. D. Zonetti, A. Saoud, A. Girard, and L. Fribourg, “A symbolic approach to voltage stability and power sharing in time-varying DC microgrids,” in *2019 18th European Control Conference (ECC)*, 2019.
13. Committee on Climate Change, “Net Zero: The UK’s Contribution to Stopping Global Warming,” 2019.
14. Energy Emergencies Executive Committee, “GB Power System Disruption–9 August 2019,” *Department for Business, Energy & Industrial Strategy, Tech. Rep*, 2019.
15. J. Figueiredo and J. Sá da Costa, “A SCADA system for energy management in intelligent buildings,” *Energy and Buildings*, vol. 49, pp. 85–98, 2012.
16. S. Nazir, S. Patel, and D. Patel, “Assessing and augmenting SCADA cyber security: A survey of techniques,” *Computers & Security*, vol. 70, pp. 436–454, 2017.
17. G. F. Franklin, J. D. Powell, A. Emami-Naeini, and J. D. Powell, *Feedback control of dynamic systems*. Prentice hall Upper Saddle River, 2002, vol. 4.
18. N. Khaled and B. Pattel, “Chapter 2 - Theoretical Foundation of MPC,” in *Practical Design and Application of Model Predictive Control*, N. Khaled and B. Pattel, Eds. Butterworth-Heinemann, 2018.
19. O. Gasparyan, *Linear and nonlinear multivariable feedback control: a classical approach*. John Wiley & Sons, 2008.
20. S. Boyd, S. P. Boyd, and L. Vandenberghe, *Convex optimization*. Cambridge university press, 2004.

21. E. F. Camacho and C. Bordons, *Introduction To Model Predictive Control*. Springer Science & Business Media, 2007.
22. J. Drgoňa, J. Arroyo, I. Cupeiro Figueroa, D. Blum, K. Arendt, D. Kim, E. P. Ollé, J. Oravec, M. Wetter, D. L. Vrabie, and L. Helsen, "All you need to know about model predictive control for buildings," *Annual Reviews in Control*, no. May, 2020.
23. S. L. Brunton and J. N. Kutz, *Data Driven Science & Engineering - Machine Learning, Dynamical Systems, and Control*. Cambridge University Press, 2017.
24. R. S. Sutton, A. G. Barto *et al.*, *Introduction to reinforcement learning*. MIT press Cambridge, 1998, vol. 135.
25. D. E. Rumelhart, G. E. Hinton, and R. J. Williams, "Learning representations by back-propagating errors," *Nature*, vol. 323, no. 6088, pp. 533–536, 1986.
26. H. Demuth, "Neural Network Toolbox Documentation," *NN with Matlab*, p. 846, 2004.
27. T. Başar and G. J. Olsder, *Dynamic noncooperative game theory*. SIAM, 1998.
28. T. Roughgarden, "Algorithmic game theory," *Communications of the ACM*, vol. 53, no. 7, pp. 78–86, 2010.
29. I. Curiel, *Cooperative game theory and applications: cooperative games arising from combinatorial optimization problems*. Springer Science & Business Media, 2013, vol. 16.
30. D. Silver, J. Schrittwieser, K. Simonyan, I. Antonoglou, A. Huang, A. Guez, T. Hubert, L. Baker, M. Lai, A. Bolton *et al.*, "Mastering the game of go without human knowledge," *nature*, vol. 550, no. 7676, pp. 354–359, 2017.
31. W. Saad, Z. Han, H. V. Poor, and T. Başar, "Game-theoretic methods for the smart grid: An overview of microgrid systems, demand-side management, and smart grid communications," *IEEE Signal Processing Magazine*, vol. 29, no. 5, pp. 86–105, 2012.
32. H. Bevrani, *Robust power system frequency control*. Springer, 2009.
33. National Grid, "The Grid Code," National Grid, Tech. Rep., 2020.
34. J. Löfberg, "Automatic robust convex programming," *Optimization Methods and Software*, vol. 27, no. 1, pp. 115–129, 2012.