Transfer Report

Keiran Suchak — 200888140

September 12, 2019

Contents

1 Introduction

A better understanding of how people move around their environment is of great utility to both academics and policy-makers. Such knowledge can be made use of in the contexts of urban planning, event management and emergency response, particularly when considering urban environments. Furthermore, this may also be of use to those interested in the social issues of mobility, inclusivity and accessibility of opportunities.

When considering such concepts, investigators often make use of modelling techniques. At their most fundamental, models represent our understanding of the system that we are studying — an understanding that may not be perfect (Stanislaw 1986). There exist modelling techniques for the simulation of how pedestrians move around urban spaces. However, these methods exist largely in isolation of the real-world — that is to say that whilst the simulations aim to reflect the real-world, there is no method by which we can incorporate up-to-date observations into these models to stop their divergence from reality.

Simulating pedestrian behaviour is often undertaken at the micro-scale, with such models typically aiming to model at the individual level or on a spatially fine-grained grid (Burstedde et al. 2001). One of the most prevalent simulation methods in this field is that of Agent-Based Modelling. Such methods consist of two key components: agents and environments. In an Agent-Based Model, we prescribe sets of rules by which individuals interact with each other and their local environments; as interactions take place on the micro-scale, we typically observe the emergence of structure at the macro-scale such as crowding (Batty et al. 2003) or lane formation (Liu et al. 2014). The evaluation of these rules is often not deterministic and instead introduces some element of randomness; these stochastic elements aim to emulate the variability of human behaviour. The introduction of such randomness in conjunction with an imperfect understanding of the

phenomena at play, however, typically result in simulation runs diverging from the real system.

In constructing their models, agent-based modellers undertake a development process that involves model verification, validation and calibration. We can take these to mean the following:

- **Model verification**: the process of ensuring that the implementation is an accurate representation of the model (Xiang et al. 2005).
- **Model validation**: the process of ensuring that the chosen model is an accurate representation of the phenomenon that we wish to study (Crooks et al. 2008).
- **Model calibration**: the process of searching for model parameter values such that we can achieve validation (Thiele et al. 2014).

Beyond this, modellers also make efforts to ensure that the initial model conditions are realistic by setting them based on historical data.

The practices of validation, calibration and setting initial model states based on historical data are appropriate for offline evaluations such as testing designs of new buildings or experimenting with different individual behaviours; however, when aiming to simulate events in real-time, this simply delays the inevitable divergence of the model from the real system. Furthermore, model parameters may be transient and thus require to be updated as time passes and the dynamics evolve.

Given the apparently inevitable divergence of stochastic simulations from the real systems that they aim to model, one may alternatively turn to big data. Data is now being generated in higher volumes and at greater velocity than ever before (Chen et al. 2014); however, there also exist issues with observation data from such systems. Whilst models typically allow us to simulate a whole system, observations are typically sparse in either time or space (or both); this is to say that observations rarely provide complete

coverage of the events. We therefore seek a solution whereby we can integrate up-to-date observations into our models as the models continue to simulate the system.

One of the methods by which we can combine knowledge represented by our model with observations as they become available is through data assimilation techniques, which are most commonly used in the field of numerical weather prediction (Kalnay 2003). Such techniques are typically made up of two steps:

1. **Predict**: Run the model forward, estimating the state of the system, until new observations become available.

2. **Update**: Upon receipt of new observations, combine the model's estimate of the system state with the new data.

These steps are repeated iteratively in a cycle. It is important to note that just as the there is error associated with the model, we also acknowledge that there is observational error associated with the data. The aim of incorporating the observations into the model is to improve the model accuracy with respect to the true system state.

A large volume of work exists in which such techniques are applied to meteorological systems where the models used are based on differential equations. Significantly less work exists in which data assimilation methods are applied to agent-based models — in particular pedestrian models. This dissertation therefore aims to expand on the pre-existing work by implementing a data assimilation scheme known as the Ensemble Kalman Filter in conjunction with a relatively simple agent-based model of pedestrians crossing a two-dimensional station from one side to the other.

1.1 Aims and Objectives

With the above in mind, this PhD research project aims to assess whether the Ensemble Kalman Filter method of data assimilation can be used to improve the accuracy with which an agent-based model can simulate pedestrian movements within urban environments. This will contribute to an emerging area of research which seeks to model social systems at close to real-time. Ultimately, the hope is that success in this research will feed into the work undertaken at Leeds City Council (with whom this project is partnered) to help with data-driven decision making processes. In order to achieve this, the following research objectives are set out:

1. Develop an implementation of the Ensemble Kalman Filter for implementation with agent-based models of pedestrian movement.

2. Evaluate the impact of filter parameters on the data assimilation method when implemented with an agent-based model.

3. Evaluate the impact of varying levels of observational information being provided to the filtering technique.

4. Explore the how varying model and agent behaviours impact filter performance.

5. Compare the performance of the Ensemble Kalman Filter with other techniques for updating agent-based models of pedestrian motion.

The remainder of this report will provide the basis for approaching the above objectives. This includes outlining a proposal for the structure of the final PhD Thesis in Section 2, reviewing existing literature in the research field in Section 3 and providing an overview of the data assimilation method around which this work centres — the Ensemble Kalman Filter — in Section 4. The report will then go on to cover the research that has been performed

so far in Section 5, before closing with a timetable for the remaining two years of the PhD and a training review in Section 6 and 7 respectively.

2 Thesis Structure

This section proposes a structures for the final output of the PhD. At present, this is expected to take the traditional form of a thesis, the structure of which is outlined in Section 2.1; the alternative option of pursuing a PhD by publication is also considered in Section 2.2. The section also addresses possibilities for other research and outputs in the form of case studies in conjunction with Leeds City Council in Section 2.3.

2.1 PhD by Thesis

Given the aims and objectives outline in Section 1.1, the following structure is proposed for the thesis.

Introduction This section will be based on Section 1, providing the background and rationale for the project, as well as outlining the main aims and objectives of the investigation.

Literature Review The foundations of this section will likely mirror Section 3, along with the addition of some coverage of:

- The use of data assimilation schemes in conjunction with agent-based models in contexts other than social simulation,
- The use of methods other than data assimilation for real-time pedestrian modelling (as well as a comparison between such approaches and data assimilation methods),

Methodology This will be based on Section 4, seeking to outline data assimilation and how it works, and will be expanded to include any other data assimilation schemes that I go on to use.

Exploring the Ensemble Kalman Filter with a simple ABM This section will be based upon the work presented in Section 5, expanding upon it by further exploring the impact of varying levels of data coverage (knowledge of all agent locations vs knowledge of a subset of agent location) and varying levels of data aggregation (knowledge of individual agent locations vs knowledge of number of agents in a given area). Furthermore, the research currently being undertaken in Section 5 assumes that the data assimilation method has knowledge regarding the origin and destination of each agent in the system; this is an unrealistic assumption, and so an investigation into the impact removing such knowledge on filter performance would also be involved. This part of the investigation would seek to understand how the different factors impact the performance of the Ensemble Kalman Filter. Furthermore, the hope is that this would act as a proof-of-concept.

Comparison of the Ensemble Kalman Filter with Different Models The research undertaken thus far has focused on implementing the Ensemble Kalman Filter for a relatively simple agent-based model — in many of the scenarios for this model, agent motion is linear and deterministic. There are likely many scenarios for which this is not the case. This section would therefore seek to apply the Ensemble Kalman Filter to different models of pedestrian motion with a view to understanding what facets of pedestrian motion the method struggles to capture.

Comparison of Different Data Assimilation Methods At present, this work has focused on the use of the Ensemble Kalman Filter in conjunction with agent-based models. There exist, however, a number of other data assimilation methods — some of which are being actively applied to the same model. This part of the investigation would seek to compare the different methods with regards to effectiveness in improving simulation accuracy, time complexity and space complexity.

Conclusion This section will draw together the research results, discussing them and how they pertain to the aims and objectives and the literature review.

2.2 PhD by Publication

The structure outlined above pertains to the traditional format of 'PhD by Thesis'. An alternative to this would be to pursue the 'PhD by Publication' route, in which the final submission would comprise of a series of papers along with and introductory section and a conclusion section. The benefits of such an approach would be that it would require that less time was set aside at the end of the PhD dedicated solely to writing the thesis, and ensuring that research is published and disseminated early in the career. The risk of this approach is that it requires that the student has one paper published, one in review and one ready to submit by the end of the PhD — papers should therefore be submitted well in advance to account for time taken on corrections and alterations. If this research were to follow such an approach, the following section may be candidates for publication:

- Exploring the Ensemble Kalman Filter with a simple ABM
- Comparison of the Ensemble Kalman Filter with Different Models
- Comparison of Different Data Assimilation Methods

The section of the methodology pertaining to the Ensemble Kalman Filter would be covered in the first publication, with any subsequent data assimilation methods being covered in the last publication.

2.3 Case Studies

Beyond the purely academic work outline so far, there is also scope to pursue external research with Leeds City Council (who are the industrial partner

for this project). Discussions regarding such case studies are ongoing, and may pertain to either of the following:

- **Pedestrian Motion on Briggate**: The previous work that has been undertaken to apply the Ensemble Kalman Filter to pedestrian agent-based models has focused on Briggate.

- **Renovation of Leeds Station**: The renovation of Leeds Station is presently ongoing. There are also plans for the redevelopment of parts of Leeds City Square. Members of Leeds City Council are therefore interested in understanding the impact of such changes, and exploring how the different potential layouts of Leeds City Square will affect the flow of pedestrians.

- **Redevelopment of the Headrow**: The redevelopment of the Headrow has recently begun whereby Leeds City Council seek to remove the central partition in the road with a view to easing the flow of public transport, better providing for cyclists, and widening pedestrian walkways and offering more green-space.

3 Literature Review

As touched upon in Section 1, the process of developing an agent-based model typically involves some form of model calibration. Model calibration is the procedure of fine-tuning the model that we are using such that it best fits the particular situation that we are seeking to model (Crooks & Heppenstall 2012). There are a large number of different manners in which we can calibrate agent-based models (Thiele et al. 2014). These approaches typically involve making use of real-world data to estimate the parameters and initial state of the model; this is, however, undertaken once prior to running the model.

In some situations, we aim to simulate events in real-time (or close to real-time). In such situations, we are often able to observe the evolution of the real-world system which we seek to model and consequently may wish to use this information to recalibrate the model. This would, however, require that we stop the simulation, undertake calibration, and restart the model. We therefore seek an approach that allows us to incorporate observations of the system whilst simulating the system — data assimilation.

As shall be seen, the application of data assimilation techniques to agent-based models has been relatively limited — this section seeks to review the existing literature regarding such application. Given the limited nature of such literature, some attention will also be given to the application of data assimilation methods to the related modelling technique of cellular automata. Consequently, the section shall proceed as follows:

- An introductory review of agent-based modelling
- A basic overview of data assimilation.
- A review of the attempts that have been made to implement data assimilation techniques to agent-based models and cellular automata.

3.1 Agent-Based Modelling

Agent-based models (sometimes referred to as individual-based models), have become widely used in a range of fields including social systems (e.g. the modelling of pedestrian movements (Liu et al. 2014)), systems pharmacology (e.g. the evolution cellular systems and the impact of different intervention measures (Cosgrove et al. 2015)), and ecological systems (e.g. the simulation of habitat-selection in fish populations (Railsback & Harvey 2002)). Whilst there have existed a number of other other modelling techniques prior to the introduction of agent-based models, many of them took the form of differential equation-based models (Parunak et al. 1998). Such equation-based models seek to describe the behaviour of systems at an aggregate level, often assuming some level of homogeneity in the population. Agent-based models, on the other hand, seek to describe the behaviour of the individuals in the system, allowing the emergence of aggregate behaviour from their interactions.

In agent-based models, individuals are characterised as discrete autonomous agents in the model which can interact with each other, as well as with their surrounding environment (Bonabeau 2002). Such a characterisation involves the model containing information for each agent regarding their attributes and the rules by which they evolve. The rules pertaining to agent interaction aim to mirror the way in which we observe interactions occurring in reality; real interactions often occur with some level of variability and so the evaluation of the respective agent rules often involve some level of randomness. A model is therefore run by iterating forward through time, evaluating agent interaction rules at each time-step.

The advantage of such an approach (and indeed other simulation approaches) when considering the research field of pedestrian movement is that it allows for the testing of scenarios that might not be possible in the physical setting of reality, may be invasive or may be expensive. Beyond this,

the benefit of using agent-based modelling techniques over equation-based methods is that they provide information to researchers regarding behaviour at an individual level, as well as the derived aggregate system behaviour.

The agent-based modelling approach is not, however, without its disadvantages. One of the most obvious of these is the computational cost of running such a model; as the scale of model increases (both in terms of the scale of the environment and the number of agents), so do the required computational resources (Bonabeau 2002). Furthermore, the stochastic nature of these models typically means that they must be run repeatedly with a given set of model parameters in order to deduce consistent system behaviour, further increasing computational cost. Such issues may be tackled through the use of supercomputing clusters and parallelisation techniques.

A further disadvantage is that such a modelling technique requires a fine-grained description of individual behaviour. This is particularly challenging when considering human social systems which can incorporate a range of complex competing motivations. Failure to correctly capture the underlying agent behaviours can result in spurious behaviours that do not match those observed in the real system (Couclelis 2002).

Calibration of Agent-Based Models

The process of calibrating an agent-based model involves identifying the values that should be assigned to model parameters in order to achieve the expected system behaviour. Such parameters govern the ways in which agent interact with each other and with their surrounding environment — it is therefore crucial that correct values be identified. In some cases, it is not possible to calculate unique values for parameters, and we instead are left with parameter sets, i.e. multiple combinations of parameters that can be used to reproduce the desired behaviour; this can result from over-paramerisation or correlation and interaction between parameters (Gan et al.

2014). The process of model calibration may also be further impeded by either a lack of data or data with high uncertainty (Thiele et al. 2014).

Associated with the process of model calibration is the procedure of sensitivity analysis. This process involves gaining an understanding of how sensitive a model is to perturbations in parameter values (Kleijnen 1995).

The remainder of this section will now go on provide a very brief overview of some of the available calibration methods; a more exhaustive review can be found in Thiele et al. (2014).

Full Factorial Design The most primitive method of model calibration is known as full factorial design, whereby a comprehensive parameter search is undertaken by running the model for every combination of parameter values (Thiele et al. 2014), evaluating for each run the outputs of the model against the desired behaviour. This method is appropriate when model runs are relatively inexpensive with regards to compute time, and when parameters take integer values; in cases where parameters take on non-integer values, care must be taken with regards to the grid resolution of the parameters that are tested as this may significantly change results.

Classical Sampling Methods Instead of the exhaustive search proposed in the form of Full Factorial Design, Classical Sampling Methods involve sampling from the set of potential parameter combinations. In its most simple form, this would take the form of randomly sampling from uniform distributions for each parameter value; however, this is ultimately very inefficient. Other sampling methods have therefore been developed.

Optimisation Methods Optimisation Methods are a common technique for fitting models. They involve defining a cost function relating some aspect of model behaviour with the corresponding aspect of the observed phenomenon; we then seek to minimise the difference (i.e. the cost). These

approaches can therefore be viewed as minimisation problems over the parameter space in which we often seek the global minimum. Many techniques exist to achieve this goal, including simulated annealing (from statistical physics) (Kirkpatrick et al. 1983) and evolutionary algorithms (Duboz et al. 2010).

Bayesian Methods Bayesian Methods seek to make use of Bayes theorem (outlined in Section 3.2) to identify appropriate parameter values (Jabot et al. 2013). This is achieved by making use of our prior understanding of the statistical distributions from which each of the parameter values are drawn; running a model a large number of times with parameter values drawn from these distributions, comparing our resulting model outputs with our observations with a view to inferring the true statistical distributions of the parameters. Due to the large number of times that models must be run, these methods can be computationally expensive; however, with the increasing levels of computational power available to researchers, they are gaining popularity (Beaumont 2010, van der Vaart et al. 2015).

3.2 Data Assimilation

The process of data assimilation involves making use of observations along with prior knowledge (which, in our case, is encoded in a model) to produce increasingly accurate estimates of variables of interest. Such a process can be achieved through a Bayesian filtering approach (Bar-Shalom et al. 2004, Jazwinski 1970).

Under such a framework, the updating of the model state is undertaken on the basis of Bayes Rule:

$$P(A|B) = \frac{P(B|A)P(A)}{P(B)} \tag{1}$$

Bayes Rule is made up of four components:

1. $P(A)$: The probability of A, known as the **Prior**.

2. $P(A|B)$: The probability of A given B, known as the **Posterior**.

3. $P(B|A)$: The probability of B given A, known as the **Likelihood**.

4. $P(B)$: The probability of B, known as the **Marginal Likelihood**.

When applying this notation to the problem at hand, the components become:

1. **Prior**, $P(\mathbf{x})$: The probability distribution representing the prior state of the model.

2. **Posterior**, $P(\mathbf{x}|\mathbf{d})$: The probability distribution representing the updated state of the model in light of the observed data, that is to say the probability of the model state given the data.

3. **Likelihood**, $P(\mathbf{d}|\mathbf{x})$: The probability distribution of the observed data given the model state.

4. **Marginal Likelihood**, $P(\mathbf{d})$: The probability distribution representing the observed data.

With the above notation, Bayes Rule becomes:

$$P(\mathbf{x}|\mathbf{d}) = \frac{P(\mathbf{d}|\mathbf{x})\,P(\mathbf{x})}{P(\mathbf{d})} \tag{2}$$

The aim of a data assimilation scheme therefore becomes to provide an update to the state in the form of the posterior, $P(\mathbf{x}|\mathbf{d})$, given new observations, $P(\mathbf{d})$.

There exist a number of different schemes for tackling this problem which are often divided into two groups (Talagrand 1997):

1. **Sequential**: Upon the arrival of a new observation, the model state is updated at the time of the new observation; includes Kalman Filter (and variations thereof), Particle Filter.

2. **Variational**: Upon the arrival of a new observation, the model solutions are updated at all times simultaneously; includes 3D-VAR, 4D-VAR.

Of the work that currently exists wherein investigators attempt to apply data assimilation schemes to agent-based models, most make use of sequential schemes.

3.3 Application of Sequential Data Assimilation to Agent-Based Models

3.3.1 Particle Filter-based Approaches

One of earliest pieces of work undertaken on the application of data assimilation schemes to agent-based models of urban environments was by Wang & Hu (2013). In this work, they simulate a smart office environment with people in it — a scenario that is becoming increasingly common with the advent of the Internet of Things (Zanella et al. 2014). The aim of the work was to make use of real-time data in conjunction with the agent-based simulation to provide more accurate estimates of the occupancy of the environment. This was achieved using the Particle Filter method of data assimilation; the method was chosen as it did not require the system to be Gaussian. The particle filter method operates by holding an ensemble of realisations of the simulation, each of which are evolved forward over time between observations; when observations are received, the particle states are weighted and the new state is obtained by sampling from these weighted particles. The observations used were synthetic data generated by the agent-based model, aiming to emulate motion sensors which would provide a binary response of whether a person was present in a given location.

The work undertaken consisted of two experiments — firstly simulating single agent in the environment, then going on to simulate two agents in the same environment. In the case if the first experiment, the agent was

simulated with two different routing behaviours; for the first routing behaviour, the agent move forward sequentially through a series of waypoints, whilst for the second routing behaviour, the agent moves through a series of waypoints before turning back to return to its initial position. In this experiment, it was found that the simulation error decreased when the agent was detected at each of the sensors for both routing behaviours with error growing between detections; this is as expected — it confirms that the simulation becomes more accurate with the addition of further information regarding the system. In the case of the second routing behaviour, the simulation error also grew following the agent's turn to head back to its origin. In the second experiment, they aimed to simulate two agents in the same environment, with the two agents maintaining spatial separation. This simulation was run a number of times for different numbers of particles with a view to establishing a relationship between the number of particles and simulation accuracy. It was found that as the number of particles was increased (through 400, 800, 1200 and 1600), the simulation error decreased. It was also found, however, that the experiments with fewer particles (400 and 800) struggled to converge, with the smaller number of particles unable to provide sufficient coverage of the state space. It was therefore noted that as the number of agents was increased, the method was likely to struggle.

This final issue may be solved by an increase in the number of particles; however, this comes with an attached increase in the computational cost (both in terms of compute time and space). The implementation of the particle filter requires that a realisation of the model be kept for each particle, resulting in growing memory requirements as the number of particles are increased. Furthermore, each particle is required to evolve the model for each time-step, resulting in an increasing computational cost.

There are two subsequent pieces of research which have sought to build directly upon the above initial investigation; each of them make use of the

same simulation model, adding different developments.

The first of these was undertaken by Rai & Hu (2013) and aimed to add a further layer by estimating behavioural patterns in the system. This was achieved using a hidden Markov model which was trained on the historical data of the system. The investigation attempts to determine the accuracy with which the model is able to identify the correct behaviour for agents, with accuracy being defined as

$$\frac{1}{T}\sum_{k=1}^{T} S_t^k - S_t^{real} \tag{3}$$

where T is taken to be the total number of simulation steps and S is the behaviour pattern state. It is unclear, however, how this calculation is undertaken given that the behavioural states are categories and not numerical; furthermore, the meaning behind the state notation is note explained — it appears that k is a time-step index, however it does not make sense to compare the behavioural state at each time state to a static "real" behavioural state and the latter would likely be a transient property. Some indication is given that a numerical encoding of the categories has been used for visualisation purposes, however this should not be used for arithmetic purposes, nor should any ordinality be inferred from it. Indeed, the results presented take the form of accuracy percentages, suggesting that a more conventional accuracy score has been used, such as

$$\frac{1}{T}\sum_{k=1}^{T} \mathbb{1}\left(\hat{S}_k = S_k^{real}\right) \tag{4}$$

where the indicator function, $\mathbb{1}(\ldots)$, returns 1 when the condition is fulfilled, else 0. The results suggest that the model developed accurately identifies the behavioural states except for states that occur infrequently; the model performs particularly well when identifying states in which agents are static, i.e. *outside* and *in conference*. Such supplementary information could improve the performance of data driven simulations, likely helping the process

of data assimilation for parameter estimation. It is worth considering, however, that the addition of this further layer to the assimilation process would also result in a further increase in the computational cost.

The second of the investigations to develop on the initial work by Wang & Hu (2013) was undertaken by Wang & Hu (2015), building more directly on the original work. This investigation made use of three different resampling schemes: standard resampling (as used in the original investigation), component sent resampling and mixed component set resampling. These different schemes were applied to a series of experiments for comparison.

The first of these experiments seeks to address the use of component set resampling. This is achieved by testing the implementation for increasing numbers of agent with component set resampling, and comparing against the corresponding results when using standard resampling. It was found that, when using standard resampling, the number of particles in the ensemble that matched with observations decreased as the number of agents increased; the use of mixed component resampling was found to reduce the rate at which this occurred. Beyond the results observed previously, this investigation also found that:

- Simulation error in a single agent system increased when an agent turned back on itself in an area where no sensors were present.
- Simulation error in a single-agent system increased when an agent approached a 3-way intersection, offering it discretely different options of direction in which to travel — the particles struggled to converge on the true state.
- Both component set resampling and mixed component set resampling improve on the accuracy gains provided by standard resampling for a multi-agent system.
- The improvements offered by mixed component set resampling over

standard resampling diminish as the number of agents increases; this was attributed to situations when agents would crowd together, thus causing difficulties for particles to distinguish different agents captured by binary sensors.

Whilst this development appears promising, the issue of computational cost (both with respect to space and time) remain largely unaddressed. The investigation focuses on small agent populations (≤ 6), for which ensemble sizes greater than 1000 are typically used to ensure that the application of data assimilation is worthwhile. This would lead to a large increase in the cost of running such experiments. When modelling more realistic systems, agent populations would likely be much larger, and consequently the number of particles required to improve accuracy would grow. This draws into question the tractability of such a method; it is likely that (at least one of) either HPC or multi-threading approaches would be required in such a situation.

3.3.2 Kalman Filter-based Approaches

Other investigations have sought to apply different data assimilation schemes including the Ensemble Kalman Filter. As shall be explained in Section 4, the Ensemble Kalman Filter is an adaptation of the original Kalman Filter (Evensen 2003). This data assimilation technique was implemented in the investigation by Ward et al. (2016), which sought to expose agent based modelling practitioners to the technique in the context of modelling how many people are in a major city at a given time. This investigation consisted of two experiments. In the first experiment, the Ensemble Kalman Filter was implemented with a simple box model that estimated how many people were present in the box based on probabilities of people entering and exiting. In the second experiment, the Ensemble Kalman Filter was applied to an epidemic-like model in which the population was split into workers and

shoppers, with works either being at home or at work, and shoppers either being susceptible to going shopping, shopping in town or having returned home after shopping.

The first experiment made use of synthetic data generated using the model with randomly drawn parameter values in order to produce ground truth data. Observations were then generated by adding normally distributed random noise to the ground truth. Running the filtering process with an ensemble of 100 realisations, data assimilation for state estimation was performed at each time-step, and it was found that on average, the error (with respect to the synthetic ground truth) of the model state was smaller after assimilation, as well as being smaller than the error in the observations. Furthermore, this approach also outperforms the theoretical steady state calculated for the system.

Beyond this, the first experiment also aimed to carry out parameter estimation by including the parameters, which are assumed to be unknown, in the state vector. In doing so, estimates of the parameters are produced at both the forecasting stage and the updating stage. The filtering process succeeds in reducing the error in the parameters with respect to the ground truth; despite this, the parameter estimates do not converge on their true values, underestimating both the arrival rate and departure rate. The ratio of the two parameters, however, is correctly estimated, suggesting that the data assimilation process has correctly estimated the governing dynamics.

The second experiment aimed to model pedestrians arriving and departing at Briggate in Leeds. Pedestrians are divided into shoppers and workers, with each group being governed by epidemic-like dynamics. Shoppers are either at home before shopping (susceptible), in town shopping (infected) or at home having returned from shopping (recovered); workers are either at home or at work. This approach seeks to more realistically represent pedestrian behaviour, designating different types of people and introducing more

complex behaviours for agents deciding to enter the city. The data used was sourced from a footfall camera on Briggate which recorded hourly counts of the number of pedestrians arriving; as such, the primary target of the assimilation process was the cumulative count of the number of agents to have arrived in the city, combining both shoppers and workers. The Ensemble Kalman Filter was applied using different ensemble sizes (10, 100 and 1000), and it was found that as the ensemble size increased, the accuracy of the simulation improved; it was noted, furthermore that the improvement observed between ensemble sizes of 10 and 100 were much greater than between ensemble sizes of 100 and 1000. Once again, parameter estimation was undertaken; however, in this case a particle filter-like approach was used.

Whilst this investigation displays that the Ensemble Kalman Filter can be implemented in conjunction with an agent-based model to successfully improve the model's prediction accuracy, it suffers from a number of shortcomings. First and foremost, it should be noted that the models used for the investigation are very simple in comparison to the majority of agent-based models; indeed, the authors admit that the models used for the investigation were not developed in the standard object-oriented framework typically used in agent-based modelling. The model used in the first experiment is a binary state model, with each agent either being in the city or not in the city. Such a model may be of use in gaining a picture of how the number of people in a city varies over time, but does not provide any additional information with regards to their spatial distribution within the city.

The inter-agent interaction governing their transition between the two states is global — this is to say that each agent's decisions are based on the state of every other agent without considering more intricate mechanisms of attraction and repulsion between agents (Helbing & Molnar 1995) such as spatially local ones. Whilst the second experiment seeks to include a richer set of behaviours by further segmenting the agents, it still fails to include any

spatial aspect, with agents again able to interact in a homogeneous fashion.

Beyond this, it should be noted that the inclusion of parameter estimation, whilst not uncommon, is not a standard approach and so some attention should be given to the impact of its inclusion in the procedure. This may be achieved by undertaking the same investigation with and without parameter estimation, comparing the accuracy with which the data assimilation scheme models the state variable of interest.

3.4 Cellular Automata

Much like agent-based models, cellular automata are a simulation method which focus on prescribing behavioural rules at a microscopic scale and allowing the emergence of macroscopic behaviour. The two approaches differ, however, in that whilst agent-based models focus on agents as the discrete indivisible units, cellular automata focus on space as the units. In such a model, the environment is divided up into a collection of spatial cells, each of which has its own set of properties. As the model is iterated forward through time, the cell properties are then updated based on rules which typically rely on the properties of a cell's neighbours. Cellular automata are often implemented in research concerning land use, in which it is common to discretise space.

Whilst this dissertation focuses on the application of data assimilation methods to agent-based models, there also exists a body of literature in which researchers make use of the same methods in conjunction with cellular automata. Specifically considering research undertaken regarding urban land use, investigations have been undertaken using both the Ensemble Kalman Filter (Li et al. 2017, Zhang et al. 2011, 2013) and the Particle Filter (Verstegen et al. 2014) This may be of interest to those concerned more broadly with the development of real-time social simulation.

3.5 Summary

As has been seen, there exist a small number of investigations which attempt to implement sequential data assimilation schemes in conjunction with agent-based pedestrian models; in each case, the filtering process lead to improvements in the modelling accuracy. Much of the work that has been undertaken has used the Particle Filter, and has made use of synthetic data; the ultimate goal of developing such a technology will inevitably be to use it with real-world data which is generated in real-time.

Each of the two data assimilation schemes that have been used have their own strengths and weaknesses. The Ensemble Kalman Filter used by Ward et al. (2016) are typically run with smaller ensemble sizes, but struggle in cases when probability distributions are non-normal. The Particle Filters used by Rai & Hu (2013), Wang & Hu (2013, 2015) resolve this issues, offering an exact solution at the cost of requiring larger ensemble sizes and consequently greater computation space and time.

This work therefore seeks to build upon the work by Ward et al. (2016) in implementing an Ensemble Kalman Filter in conjunction with a agent-based pedestrian model.

4 Method

There exist a number of different data assimilation schemes, many of which are used extensively in fields such as numerical weather prediction, navigation and tracking, and other forecasting problems. Such methods, however, have been sparsely used in the field of real-time urban simulation and as such this investigation attempts to build upon the existing work by implementing the Ensemble Kalman Filter in conjunction with a pedestrian agent-based model. This chapter therefore seeks to outline the method used in this investigation — the Ensemble Kalman Filter. As shall be explained, the Ensemble Kalman Filter is an approximation of the Kalman Filter, and as such some attention will first be given to the original Kalman Filter, followed by an explanation of the Ensemble Kalman Filter along with the innovations that it incorporates.

4.1 Kalman Filter

One of the earliest forms of Bayesian filtering is the sequential data assimilation scheme known as the Kalman Filter, which forms the foundation of this piece of work. As with other sequential data assimilation schemes, the Kalman Filter operates on a model with a given state and forecasting process, updating the state and associated covariance matrix upon receipt of new observations. This is undertaken as part of the predict-update cycle. The prediction process is undertaken by applying the modelling operator to both the model state and model state covariance. The update process is undertaken based on the uncertainty in the model forecasts and the observation uncertainty with a view to minimising the posterior mean squared error. Under the Bayesian framework outlined in Section 3.2, we refer to the model state vector and associated covariance before updating as the prior, $\mathbf{x}$ and $\mathbf{Q}$, and to the model state and associated covariance after updating as the posterior, $\hat{\mathbf{x}}$ and $\hat{\mathbf{Q}}$. Given new observations, $\mathbf{d}$, the posterior state

vector is given by

$$\hat{\mathbf{x}} = \mathbf{x} + \mathbf{K}\left(\mathbf{d} - \mathbf{H}\mathbf{x}\right), \tag{5}$$

and the posterior covariance is given by

$$\hat{\mathbf{Q}} = \left(\mathbf{I} - \mathbf{K}\mathbf{H}\right)\mathbf{Q}, \tag{6}$$

where $\mathbf{K}$ is the Kalman gain matrix and $\mathbf{H}$ is the observation operator. The Kalman gain matrix is given by

$$\mathbf{K} = \mathbf{Q}\mathbf{H}^T\left(\mathbf{H}\mathbf{Q}\mathbf{H}^T + \mathbf{R}\right)^{-1} \tag{7}$$

where $\mathbf{R}$ is the observation covariance.

In the case when the errors are normally distributed, this filter provides an exact posterior estimate. However, this approach suffers from two issues. The first of these is that it assumes that the operators that are applied (namely the model transition operator and the observation operator) are linear; this is often not the case with more complex systems, particularly when the system elements interact with each other (as is typically the case in agent-based models). Furthermore, as the dimensionality of the model increases, the cost of propagating forward the model state covariance may increase to the point where it is intractable.

A number of approaches have been developed which attempt to solve these problems, one of which is the Ensemble Kalman Filter.

4.2 Ensemble Kalman Filter

In order to address some of these problems, the Ensemble Kalman Filter was developed (Evensen 2003, 2009), which acts as an approximation of the Kalman Filter. This approximation is achieved through a Monte Carlo approach of using an ensemble of sample state vectors to represent the state distribution; this development mirrors the recent incorporation of Monte Carlo methods in the field of Bayesian statistics (Wikle & Berliner 2007).

The remainder of this section will seek to outline this framework in a manner similar to that documented by Mandel (2009). The state is represented as an ensemble of state vectors as follows:

$$\mathbf{X} = [\mathbf{x}_1, \ldots, \mathbf{x}_N] = [\mathbf{x}_i], \quad \forall i \in (1, N), \tag{8}$$

where the state ensemble matrix, $\mathbf{X}$, consists of N state vectors, $\mathbf{x}_i$. The mean state vector, $\bar{\mathbf{x}}$, can be found by averaging over the ensemble:

$$\bar{\mathbf{x}} = \sum_{i=1}^{N} \mathbf{x}_i. \tag{9}$$

Similarly, the observations are represented as follows:

$$\mathbf{D} = [\mathbf{d}_1, \ldots, \mathbf{d}_N] = [\mathbf{d}_i], \quad \forall i \in (1, N), \tag{10}$$

with each member of the data ensemble matrix, $\mathbf{D}$, being the sum of the original observation $\mathbf{d}$, and a random vector, ϵ_i:

$$\mathbf{d}_i = \mathbf{d} + \epsilon_i, \quad \forall i \in (1, N). \tag{11}$$

The random vector is drawn from an unbiased normal distribution:

$$\epsilon \sim \mathcal{N}(0, \mathbf{R}).$$

As with the model state, the mean data vector, $\bar{\mathbf{d}}$, can be found by averaging over the ensemble:

$$\bar{\mathbf{d}} = \sum_{i=1}^{N} \mathbf{d}_i. \tag{12}$$

Given that the noise added to the original data vector is unbiased, we should expect that the mean data vector converges to the original data vector, $\mathbf{d}$:

$$\lim_{N \to \infty} \bar{\mathbf{d}} = \mathbf{d}.$$

By implementing these adaptations, the Ensemble Kalman Filter aims to address the issues faced by the original Kalman Filter with respect to forecasting the state covariance matrix; more specifically, as a result of this

approach the state covariance matrix no longer needs to be forecast by applying the model operator, but instead can simply be generated as a sampling covariance. Consequently, concerns over the computational cost of forecasting the covariance matrix and over the requirement that the forecasting process be undertaken by applying a linear operator to the covariance matrix are greatly reduced.

Given the above framework, the data assimilation is once again made up of the predict-update cycle, with the updating of the state ensemble, $\hat{\mathbf{X}}$, begin undertaken on the basis of the following equation:

$$\hat{\mathbf{X}} = \mathbf{X} + \mathbf{K}\left(\mathbf{D} - \mathbf{H}\mathbf{X}\right), \tag{13}$$

where $\mathbf{H}$ is once again the observation operator. In this case, the Kalman gain matrix, $\mathbf{K}$ is given by

$$\mathbf{K} = \mathbf{C}\mathbf{H}^T\left(\mathbf{H}\mathbf{C}\mathbf{H}^T + \mathbf{R}\right)^{-1}. \tag{14}$$

in which the previous state covariance matrix, $\mathbf{Q}$, has been replace with the sample state covariance matrix, $\mathbf{C}$.

5 Undertaken Research

As outlined in Chapter 4, the Ensemble Kalman Filter is a data assimilation method which aims to approximate the effect of the original Kalman Filter by representing the state of the model on which it is operating as an ensemble of state samples. The aim of applying this ensemble method is to overcome the difficulties encountered by the original Kalman Filter when being applied to non-linear models, and when attempting to propagate the state covariance matrix for models with high dimensionality. This method has been applied to an agent based model of pedestrian movement (documented in Section 5.1) with a view to reducing the error in the model with respect to the ground truth. It should be noted that this section adopts the common terminology of "forecast" to mean the state predicted by the model before updating, and "analysis" to mean the state after updating.

This section will present the results of the initial experiments that have been undertaken. These experiments seek to show that the Ensemble Kalman Filter is effective in reducing simulation error when implemented for a simple agent-based model, and furthermore seeks to explore the impact of different factors on this error reduction. This will be achieved by first showing the impact of the filtering process on the model state, in particular focussing on a single agent in the model. It will then be showed how the simulation error varies over time, comparing the evolution of the error in the forecast, the error in the analysis and the error in the observations. Finally, results will be presented for the exploration of the impact of varying ensemble size, assimilation period, population size and measurement error on the effectiveness of the filter.

5.1 Model

The Ensemble Kalman Filter outlined in Section 4.2 is implemented in conjunction with a pedestrian agent-based model known as `StationSim`[1] which aims to simulate pedestrians crossing from one side of the station to the other. StationSim has been developed under an object-oriented framework, and as such both the model and the agent are represented by classes. The state of each agent, a_i, comprises of its positions in two-dimensional continuous space:

$$a_i = (x_i, y_i)\,. \tag{15}$$

The state of the model, m, comprises of the collection of all of the agents in the model:

$$\begin{aligned} m &= [a_0, \ldots, a_N]^T \\ &= [x_0, y_0, \ldots, x_N, y_N]^T\,, \end{aligned} \tag{16}$$

where N is the population size. At each time-step, the model state is evolved by sequentially evolving the agents.

The model is initialised by passing a number of arguments to the constructor, including the number of agents in the population and the dimensions of the rectangular environment. Upon initialisation, the model generates a population of agents, each of which are randomly allocated the following:

- Entrance through which to enter the environment.
- Exit through which to leave the environment.
- Speed at which to traverse the environment.

As shown in Figure 1, entrances are located on the left-hand side of the rectangular environment, and exits are located on the right-hand side of

[1] https://github.com/Urban-Analytics/dust/blob/master/Projects/ABM_DA/stationsim/stationsim_model.py

the environment, with each agent seeking to traverse the environment from their respective entrance to their respective exit. Agents' motion across the environment is largely linear until they interact with each other. Where the paths of agents intersect in time and space, crowding occurs. Faster agents attempt to pass slower agents, at times getting stuck behind them — this can be observed in the trails show in Figure 1 at $(x, y) \approx (65, 55)$.

The above phenomenon is a consequence of the agent behaviour outlined in Figure 2. In Figure 2 (and subsequently Figure 3), the `status` variable refers to whether an agent is active or not; a `status` of 0 indicates that the agent is not active but is waiting to become active, a `status` of 1 indicates that the agent is active and is traversing the system, a `status` of 2 indicates that the agent has left the system and is no longer active. This variable controls agents' activation and deactivation behaviours outlined in Figure 3. The crowding phenomenon occurs due to the behaviour described in Figure 4, whereby agents attempt to move as far directly towards their target destination as possible. If they are unable to proceed forward, they engage the `wiggle` behaviour where by they choose to move up or down (in the y-direction) in an attempt to pass the obstruction in front of them; the decision to move either up or down is made randomly with each choice carrying an equal probability.

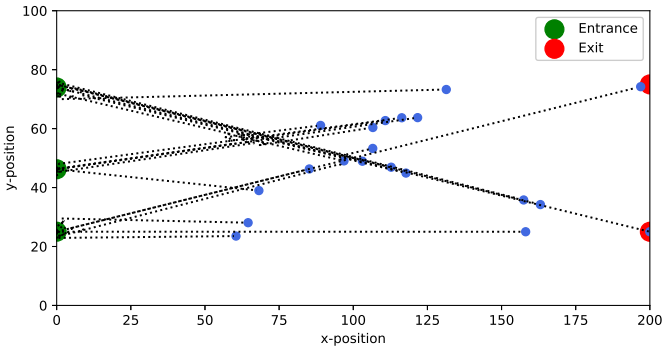


Figure 1: Sample model run.

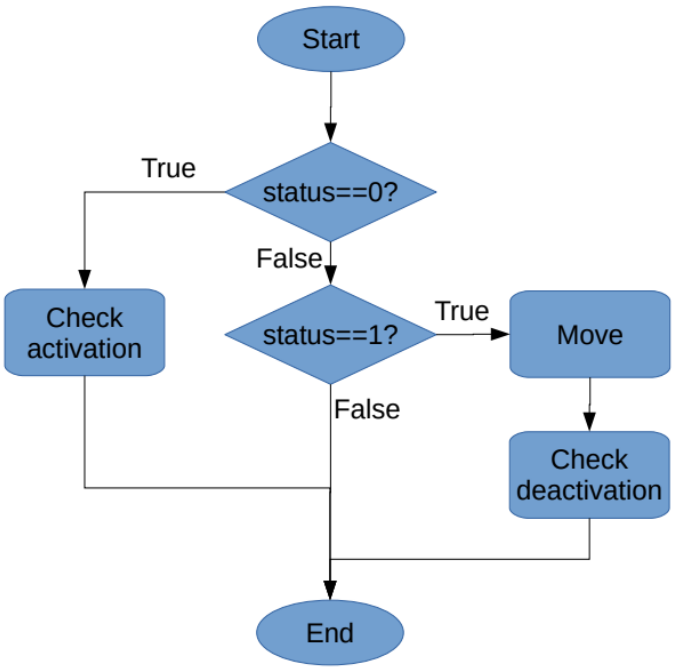
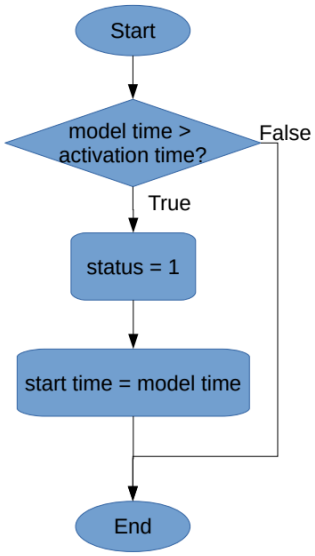


Figure 2: Flow diagram of agent step behaviour. Activation and deactivation behaviours are defined in Figure 3; movement behaviour is defined in Figure 4.

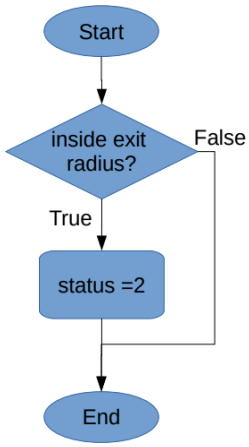
5.2 Experimental Setup

With the above model in mind, a set of experiments were designed. This first involved confirming that the Ensemble Kalman Filter was effective in reducing the simulation error. We then went on to explore the way in which different factors impact the performance of the Ensemble Kalman Filter when applied to an agent-based model. In particular, the factors that were explored were:

- **Ensemble Size**: The number of copies of the model that the filter maintains.
- **Assimilation Period**: The number of time steps between each suc-



(a) Activation



(b) Deactivation

Figure 3: Flow diagrams of agent behaviours for activation and deactivation.

cessive observation being used to update the model states.

- **Measurement Error**: The standard deviation of the noise applied to the ground truth data in order to obtain observations.
- **Population Size**: The number of agents created in the model.

In order to undertake this investigation, a class was developed in Python to represent the Ensemble Kalman Filter. The Python class representing the model is passed to the filter class as an argument, along with the filter ensemble size, the frequency with which the filter should assimilate data, and parameters governing the observational noise. Upon construction, an instance of the filter class creates an instance of the model, referred to as the `base_model`, which takes the parameters given in Table 4. Subsequently, an ensemble of models is created by taking deep-copies of the `base_model`, thus ensuring that each of the ensemble members starts with model- and agent-attributes that match those in the `base_model`.

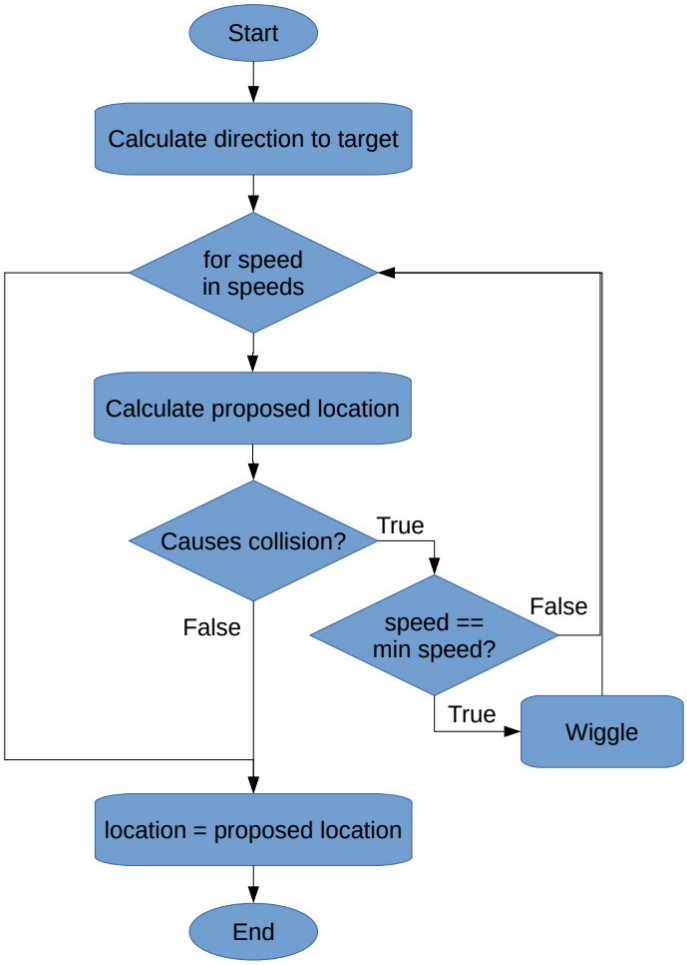


Figure 4: Flow diagram of agent movement behaviour.

The instance of the filter class steps forward through time according to the predict-update cycle. At each predict step, each of the ensemble member models are stepped forward once, along with the base model; the frequency with which the filter undertakes the update step is governed by a parameter known as the `assimilation_period`. Upon reaching the update step of each cycle, the state of each of the ensemble member models is updated according to equations found in Section 4.2. The ground truth for these experiments is taken to be synthetic data generated by the `base_model`; observations are

then generated by adding normally distributed random noise to the ground truth at each assimilation step. Errors are calculated as the root-mean-squared error averaging over the population of agents:

$$RMSE = \sqrt{\frac{1}{N}\sum_{i=0}^{N}(\hat{x}_i - x_i)^2},$$

where $\hat{x}_i$ is the model state of the ith agent, x_i is the ground truth state of the ith agent and N is the population size.

Experiment 1: Confirming that Ensemble Kalman Filter is effective This experiment aims to examine the impact of the Ensemble Kalman Filter on the model state through the visual examination of the model state before and after the update process. This is achieved by running the Ensemble Kalman Filter implementation with the filter and model parameters found in Tables 1 and 2 respectively. The mean model state will be plotted for one of the agents in the model, along with each of the positions for the agent in each of the ensemble of models and the observed position, both before the update and after the update. The expectation is that the update process reduces the uncertainty in the model's estimate of the agent's position, and that the posterior estimate is more accurate than the prior estimate.

Experiment 2: Exploring evolution of errors over time This experiments aim to compare the forecast error, the analysis error and the observation error of the system, and examine how they each vary over time. In order to achieve this, 10 realisations of the system are run using the same filter and model parameters found in Tables 1 and 2. The forecast, analysis and observation errors are extracted from each of these realisations at each point in model time when the Ensemble Kalman Filter updates the states of each of the models in the ensemble (i.e. every 50 time-steps). The each

set of errors are averaged over the realisations with a view to finding the average behaviour for each and comparing how they vary over model time.

Experiment 3: Exploring the impact of filter parameters The third and final experiment seeks to explore the impact of variations in filter parameters on filter performance — in particular the impact of assimilation period, ensemble size and measurement error (as outlined above). This is achieved by running the system for a variety of parameter values shown in Table 3, and analysing how the behaviour of the system changes with an increasing population size. For each combination of parameters, the system is run 10 times, each time extracting the forecast, analysis and observation errors for each point in model time at which the system is updated with observation data. Outputs are averaged over both the realisations and the simulation time to produce a individual forecast, analysis and observation error values for each of the parameter combinations. These are plotted as heatmaps, focusing on the variation in analysis error with population size and the filter parameter of choice. As the population size increases, the number of interactions between agents increases, leading to more random decisions being made as a consequence of the `wiggle` behaviour. As a result, the predicted model states are likely to deviate further from the ground truth. It is expected that increasing the filter ensemble size will reduce this impact, as should reducing the assimilation period.

Variable	Value
Number of iterations	300
Assimilation period	50
Ensemble size	10
Observation standard deviation	1.0

Table 1: Table of filter parameters for experiments 1 and 2.

Variable	Value
Population size	20
Number of entrances	3
Number of exits	2
Environment height	100
Environment width	200

Table 2: Table of model parameters for experiments 1 and 2.

Variable	Value
Number of iterations	300
Assimilation period	$[2, 5, 10, 20, 50]$
Ensemble size	$[2, 5, 10, 20, 50]$
Observation standard deviation	$[0.5, 1.0, 1.5, 2.0, 2.5]$

Table 3: Table of filter parameters for experiments.

Variable	Value
Population size	$[5, 10, 15, 20, 25]$
Number of entrances	3
Number of exits	2
Environment height	100
Environment width	200

Table 4: Table of model parameters for experiment.

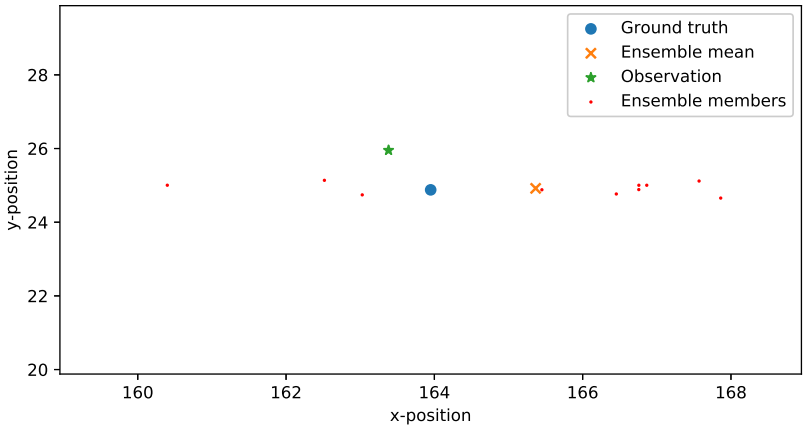
5.3 Results

Experiment 1: Confirming that Ensemble Kalman Filter is effective Figure 5 shows the impact of the applying the Ensemble Kalman Filter on one of the agents in the model. The observation error is considered as the distance between the observed agent state and the true agent state — this remains unchanged between Figures 5a and 5b; the forecast error is the distance between the ensemble mean agent state and the ground truth agent state in Figure 5a; the analysis error is the distance between the ensemble mean agent state and the ground truth agent state in Figure 5b.

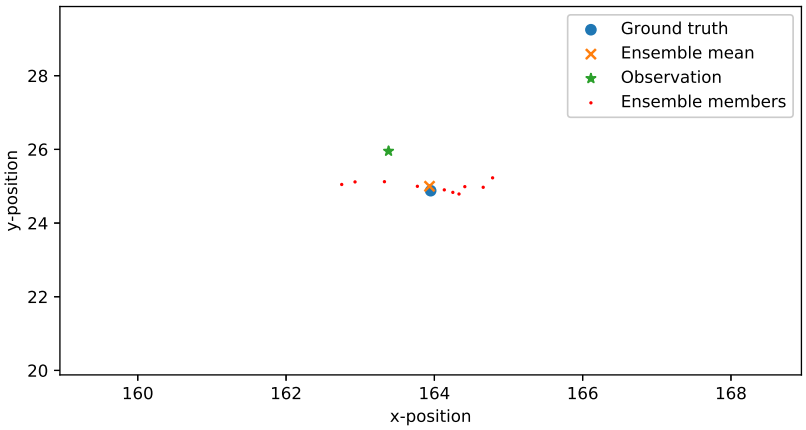
It can be seen by in Figure 5a that the forecast error is greater than the observation error. Furthermore, it can be seen that the variance in the forecast error for the ensemble members is large with some members lying as close to the ground truth as the observed agent state and others lying approximately fours times as far away. The majority of the variation occurs in the x-direction, indicating that in the case of some ensemble member models the agent in question likely became involved in some crowding and became stuck behind other agents, whilst in other ensemble member models the agent may be ahead of the crowd and has proceeded unimpeded.

Comparing Figures 5a and 5b reveals two points. The first is that ensemble mean agent state lies much closer to the ground truth agent state — closer than the observed agent state. Indeed, the improvement in accuracy is such that the ensemble mean agent state at almost exactly the same location as the ground truth agent state. Beyond this, it can be seen that the variance in the error for the ensemble members has greatly reduced. These two observations suggest that the Ensemble Kalman Filter is effective in improving the accuracy with which the model can simulate the system.

To ensure that this result is not restricted to the individual agent, the next experiment will go on to average the errors over the whole agent population.



(a) Before



(b) After

Figure 5: Comparison of model state before and after updating via the Ensemble Kalman Filter, focusing on a single agent in the system.

Experiment 2: Exploring evolution of errors over time This experiment seeks to consider the variations in error over time. In particular Figure 6 consists of three sub-plot, in each case showing the evolution of average error per agent over time. The experiment is run for 10 realisations, each time using the same model and filter parameters. Each dashed line in Figure 6 is indicative of a single realisation, whilst the solid line is the

average over all realisations.

Trivially, it can be observed that the average observation error remains approximately constant over all model time, with very little variation between different realisations. This is reflective of the standard deviation of the normally distributed random noise that was added to the ground truth in order to produce the observations. Beyond this, it can also be seen that the forecast and analysis error at the very beginning of model time are negligible. This is a consequence of how the ensemble of models is constructed; the ensemble members are constructed by taking deep-copies of the model which produces the ground truth and therefore at $t = 0$ the agents in each of the models is exactly the same as in the ground truth model.

As model time progresses, and we approach $t = 50$ (i.e. the first assimilaton point), we observe that the mean forecast error has grown to appoximately 2.2. This is likely a result of many of the agents entering the system over the first 50 time-steps, likely resulting in some crowding. The application of the assimilation process results in a reduction in the mean error, as seen through the comparison of the mean forecast error and mean analysis error at $t = 50$. The same can be seen at each of the subsequent assimilation points, with the mean analysis error being lower than the mean forecast error each time.

It is noticeable that there is variation between individual realisations. In the worst-case scenario, i.e. the uppermost dashed lines for both the forecast and the analysis, it can be seen that the mean analysis error is approximately as good as the mean forecast error if not better for the majority of assimilation points.

Experiment 3: Exploring the impact of filter parameters This experiment seeks to explore the impact on analysis error of variations in filter parameters, and how this behaviour changes over different population

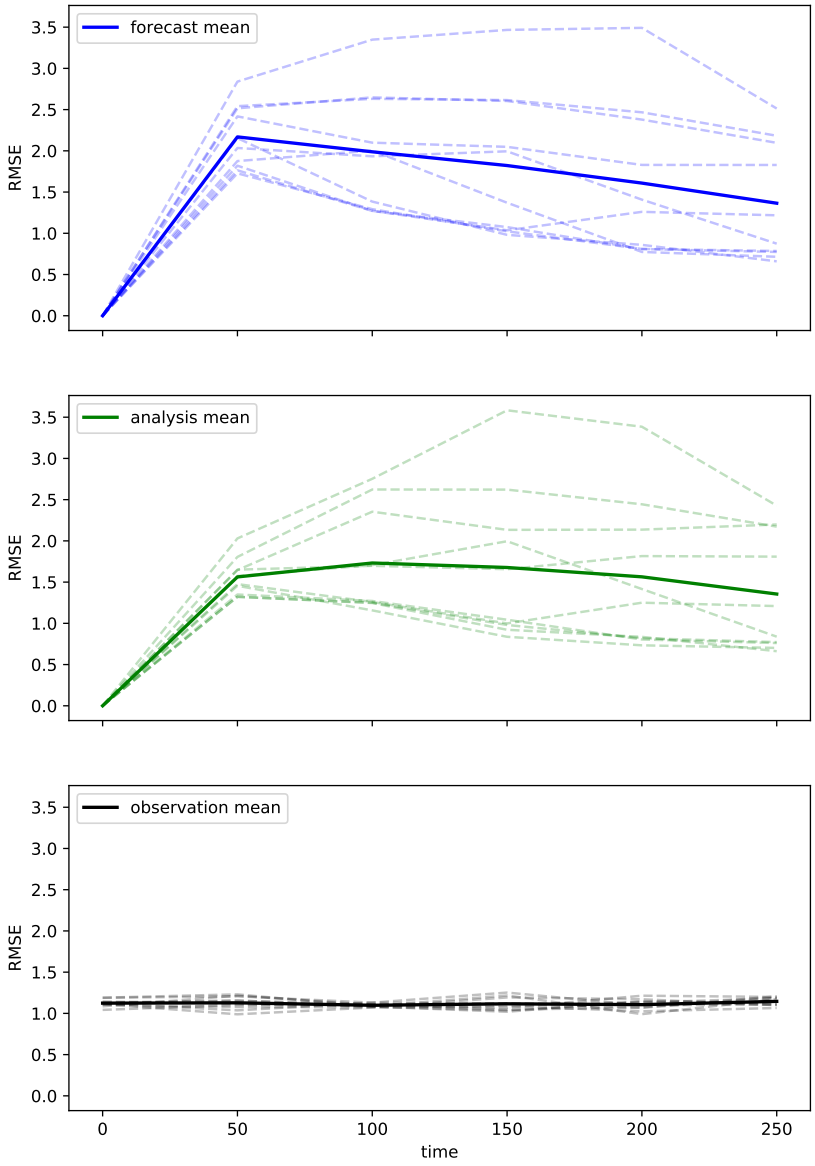


Figure 6: Evolution of forecast error, analysis error and observation error over model time; solid lines indicate mean, dashed lines indicate individual realisations.

sizes. This information is captured in Figures 7, 8, 9. These figures remove the temporal component seen in the previous experiment from the analysis, averaging over each of the assimilation points. In the case of each of the figures, darker shades indicate higher mean analysis error, whilst lighter shades indicate lower mean analysis error.

Considering first Figure 7, it can be seen that as population size increases so does the mean analysis error. This is expected — an increase in the model population size results in more crowding and therefore the introduction of more random decisions made by agents. Conversely, a reduction in the model population size leads to few stochastic elements being introduced, resulting in the model becoming more linear and deterministic. Beyond this, it can be seen that as ensemble size increases, mean analysis error decreases. Once again, this is expected — as the ensemble size becomes infinitely large, the Ensemble Kalman Filter is expected to converge on the original Kalman Filter, providing the optimum solution for the Bayesian updating problem of estimating the posterior (i.e. analysis) model state.

Considering Figure 8, it can once again be seen that as population size increases, so does the mean analysis error. The figure also suggests that as assimilation period increases, there may be a slight reduction in mean analysis error. This is contrary to expectation — it was expected that a lower assimilation period would result in improvements in mean analysis error as more frequently assimilating observations into the model would help to keep the ensemble members closer to the ground truth. It it worth noting, however, that the trend with respect to assimilation period only appears to be slight and appears to fade as the population size increases. This phenomenon may warrant further exploration with a greater number of realisations.

Considering Figure 9, it can once again be seen that as population size increases, so does thee mean analysis error. The figure also suggests that

as the standard deviation of the measurement error decreases, the mean analysis error also decreases. This highlights the value of high fidelity observations. With lower uncertainty observations, the Kalman gain matrix places a greater weighting on the perturbation to the model state provided by the observed data.

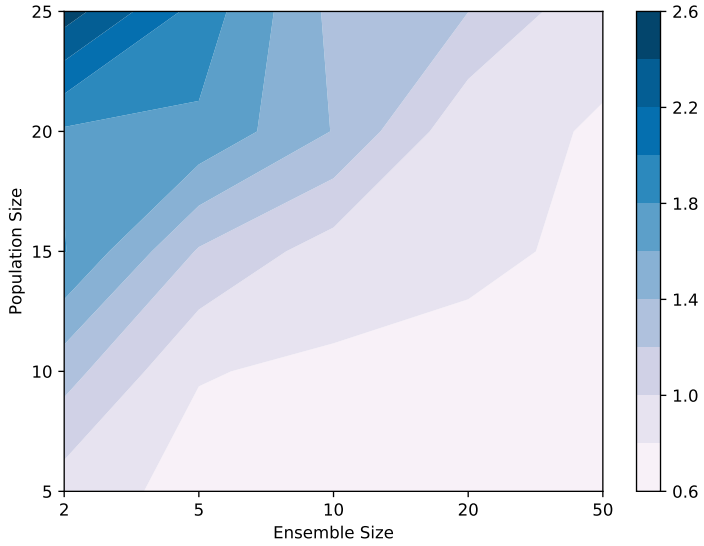


Figure 7: Variation of mean analysis error with filter ensemble size and model population size; fixed assimilation period (20), fixed measurement error standard deviation (1.5).

5.4 Conclusion

The work presented in this section aims to form the basis of the remaining PhD research. It has involved the development of an Ensemble Kalman Filter for implementation in conjunction with agent-based model. Preliminary experiments have shown the filter to be effective in improving the accuracy with which the provided agent-based model can model pedestrian motion. It has also been shown that, as expected, an increase in ensemble size has resulted in improved filter performance. There remain, however, a number

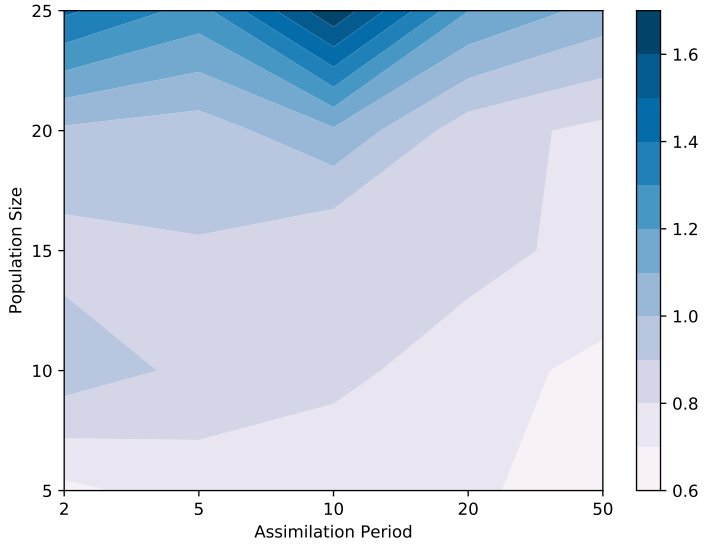


Figure 8: Variation of mean analysis error with filter assimilation period and model population size; fixed ensemble size (20), fixed measurement error standard deviation (1.5).

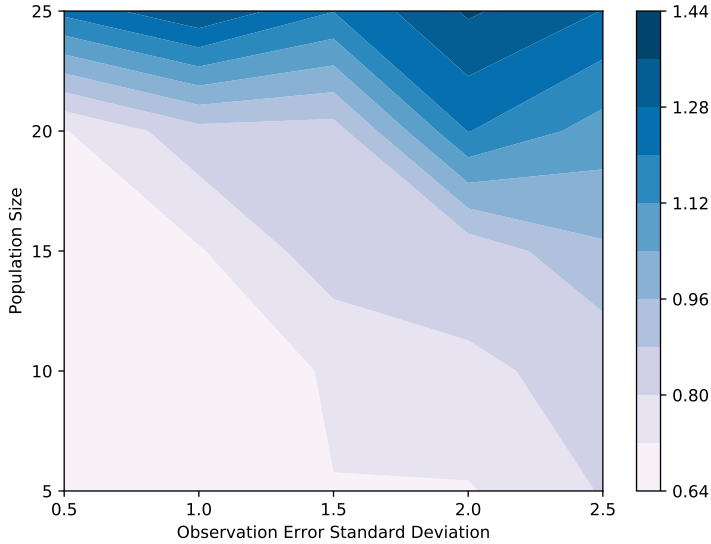


Figure 9: Variation of mean analysis error with measurement error standard deviation and model population size; fixed ensemble size (20), fixed assimilation period (20).

of avenues that require further investigation; these include a further exploration of the impact of varying assimilation period on filter performance, as well as the objectives outlined in Section 2.

6 Research Timetable

This section aims to outline how the remaining time in the PhD will be allocated. In Section 2, the structure for the PhD Thesis was proposed. This structure lends itself to three phases of research, each pertaining to an objective:

- Objective 1: exploring the Ensemble Kalman Filter with a simple ABM.
- Objective 2: comparing the effectiveness of the Ensemble Kalman Filter when applied to different models.
- Objective 3: comparing the effectiveness of the Ensemble Kalman Filter with other data assimilation methods.

These objectives are timetabled as shown in the Gantt chart in Figure 10. The Gantt chart also proposes some of the conference that may be attended to present parts of the research; GISRUK in the Spring of 2020 would provide the opportunity to present the results of Objective 1, SIMSOC in the Autumn of 2020 would provide the opportunity to present the results of Objective 2 and SIMSOC in the Autumn of 2021 would provide the opportunity to present the results of Objective 3. Subsequent RSG meetings are also expected to fall approximately around the end of each of the Objectives.

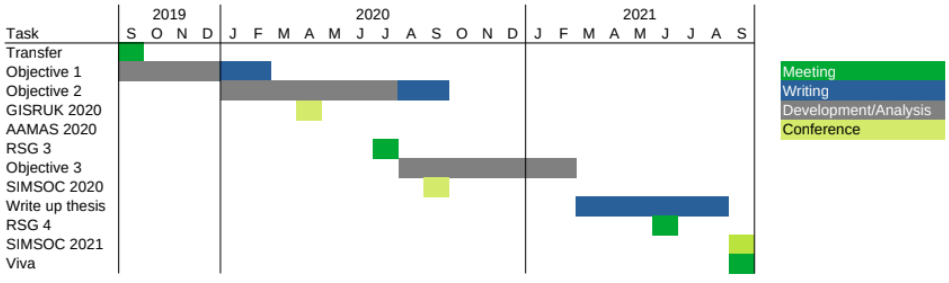


Figure 10: Gantt chart outline allocation of time over next two years.

7 Training Plan Review

7.1 Training Provided

The training provided thus far has take two form — technical training undertaken as taught modules on the CDT programme and non-technical training pursued via other avenues. Aside from the core set of modules undertaken for the CDT, I have also undertaken the following option modules:

- Big Data Consumer Analytics
- Parallel and Concurrent Programming (COMP5811M)
- Programming for Geography Information Analysis: Advanced Skills (GEG5790M)

These have aimed to gain a better understanding data scientific tools and to improve my programming knowledge. The other non-academic forms have training that I have undertaken are as follows:

- Foundations in Teaching (University of Leeds) — Provided with an introduction to teaching and demonstrating as a PGR.
- Identifying Impact Goals (University of Leeds) — Learned about reframing research with a view to achieving impact.
- Networking (University of Leeds) — Gain skills for professional networking at conferences.
- Time Management (University of Leeds) — Reviewed methods for time management.
- Literature Searching (University of Leeds) — Learned about how to undertake more thorough and systematic literature reviews.

- Facilitator training (Turing Institute Data Study Group) — Gained interpersonal skills and a greater understanding of how to manage a research project.

7.2 Training Required

The training provided thus far has been substantial; consequently, comparatively little time will be spent seeking training, with more time being allocated to PhD research. The training that shall be sought, however, will focus on writing:

- Writing for Academic Publication — The CDT is hosting a workshop on getting work published at its annual event this month which I plan to attend.
- Thesis Writing — The ODPL offers a series of workshops on developing writing skills specifically focused on thesis writing.

References

Bar-Shalom, Y., Li, X. R. & Kirubarajan, T. (2004), *Estimation with applications to tracking and navigation: theory algorithms and software*, John Wiley & Sons.

Batty, M., DeSyllas, J. & Duxbury, E. (2003), 'The discrete dynamics of small-scale spatial events: agent-based models of mobility in carnivals and street parades', *International Journal of Geographical Information Science* **17**(7), 673–697.

Beaumont, M. A. (2010), 'Approximate bayesian computation in evolution and ecology', *Annual review of ecology, evolution, and systematics* **41**, 379–406.

Bonabeau, E. (2002), 'Agent-based modeling: Methods and techniques for simulating human systems', *Proceedings of the national academy of sciences* **99**(suppl 3), 7280–7287.

Burstedde, C., Klauck, K., Schadschneider, A. & Zittartz, J. (2001), 'Simulation of pedestrian dynamics using a two-dimensional cellular automaton', *Physica A: Statistical Mechanics and its Applications* **295**(3-4), 507–525.

Chen, M., Mao, S. & Liu, Y. (2014), 'Big data: A survey', *Mobile networks and applications* **19**(2), 171–209.

Cosgrove, J., Butler, J., Alden, K., Read, M., Kumar, V., Cucurull-Sanchez, L., Timmis, J. & Coles, M. (2015), 'Agent-based modeling in systems pharmacology', *CPT: pharmacometrics & systems pharmacology* **4**(11), 615–629.

Couclelis, H. (2002), 'Modeling frameworks, paradigms, and approaches', *Geographic Information Systems and Environmental Modelling, Prentice Hall, London* .

Crooks, A., Castle, C. & Batty, M. (2008), 'Key challenges in agent-based modelling for geo-spatial simulation', *Computers, Environment and Urban Systems* **32**(6), 417–430.

Crooks, A. T. & Heppenstall, A. J. (2012), Introduction to agent-based modelling, *in* 'Agent-based models of geographical systems', Springer, pp. 85–105.

Duboz, R., Versmisse, D., Travers, M., Ramat, E. & Shin, Y.-J. (2010), 'Application of an evolutionary algorithm to the inverse parameter estimation of an individual-based model', *Ecological modelling* **221**(5), 840–849.

Evensen, G. (2003), 'The ensemble kalman filter: Theoretical formulation and practical implementation', *Ocean dynamics* **53**(4), 343–367.

Evensen, G. (2009), 'The ensemble kalman filter for combined state and parameter estimation', *IEEE Control Systems Magazine* **29**(3), 83–104.

Gan, Y., Duan, Q., Gong, W., Tong, C., Sun, Y., Chu, W., Ye, A., Miao, C. & Di, Z. (2014), 'A comprehensive evaluation of various sensitivity analysis methods: A case study with a hydrological model', *Environmental Modelling & Software* **51**, 269–285.

Helbing, D. & Molnar, P. (1995), 'Social force model for pedestrian dynamics', *Physical review E* **51**(5), 4282.

Jabot, F., Faure, T. & Dumoulin, N. (2013), 'Easy abc: performing efficient approximate b ayesian computation sampling schemes using r', *Methods in Ecology and Evolution* **4**(7), 684–687.

Jazwinski, A. H. (1970), 'Mathematics in science and engineering', *Stochastic processes and filtering theory* **64**.

Kalnay, E. (2003), *Atmospheric modeling, data assimilation and predictability*, Cambridge university press.

Kirkpatrick, S., Gelatt, C. D. & Vecchi, M. P. (1983), 'Optimization by simulated annealing', *science* **220**(4598), 671–680.

Kleijnen, J. P. (1995), 'Sensitivity analysis and related analyses: a survey of statistical techniques'.

Li, X., Lu, H., Zhou, Y., Hu, T., Liang, L., Liu, X., Hu, G. & Yu, L. (2017), 'Exploring the performance of spatio-temporal assimilation in an urban cellular automata model', *International Journal of Geographical Information Science* **31**(11), 2195–2215.

Liu, S., Lo, S., Ma, J. & Wang, W. (2014), 'An agent-based microscopic pedestrian flow simulation model for pedestrian traffic problems', *IEEE Transactions on Intelligent Transportation Systems* **15**(3), 992–1001.

Mandel, J. (2009), 'A brief tutorial on the ensemble kalman filter', *arXiv preprint arXiv:0901.3725* .

Parunak, H. V. D., Savit, R. & Riolo, R. L. (1998), Agent-based modeling vs. equation-based modeling: A case study and users' guide, *in* 'International Workshop on Multi-Agent Systems and Agent-Based Simulation', Springer, pp. 10–25.

Rai, S. & Hu, X. (2013), Behavior pattern detection for data assimilation in agent-based simulation of smart environments, *in* '2013 IEEE/WIC/ACM International Joint Conferences on Web Intelligence (WI) and Intelligent Agent Technologies (IAT)', Vol. 2, IEEE, pp. 171–178.

Railsback, S. F. & Harvey, B. C. (2002), 'Analysis of habitat-selection rules using an individual-based model', *Ecology* pp. 1817–1830.

Stanislaw, H. (1986), 'Tests of computer simulation validity: what do they measure?', *Simulation & Games* **17**(2), 173–191.

Talagrand, O. (1997), 'Assimilation of observations, an introduction', *Journal of the Meteorological Society of Japan* **75**(1B), 191–209.

Thiele, J. C., Kurth, W. & Grimm, V. (2014), 'Facilitating parameter estimation and sensitivity analysis of agent-based models: A cookbook using netlogo and r', *Journal of Artificial Societies and Social Simulation* **17**(3), 11.

van der Vaart, E., Beaumont, M. A., Johnston, A. S. & Sibly, R. M. (2015), 'Calibration and evaluation of individual-based models using approximate bayesian computation', *Ecological Modelling* **312**, 182–190.

Verstegen, J. A., Karssenberg, D., Van Der Hilst, F. & Faaij, A. P. (2014), 'Identifying a land use change cellular automaton by bayesian data assimilation', *Environmental Modelling & Software* **53**, 121–136.

Wang, M. & Hu, X. (2013), Data assimilation in agent based simulation of smart environment, *in* 'Proceedings of the 1st ACM SIGSIM Conference on Principles of Advanced Discrete Simulation', ACM, pp. 379–384.

Wang, M. & Hu, X. (2015), 'Data assimilation in agent based simulation of smart environments using particle filters', *Simulation Modelling Practice and Theory* **56**, 36–54.

Ward, J. A., Evans, A. J. & Malleson, N. S. (2016), 'Dynamic calibration of agent-based models using data assimilation', *Royal Society open science* **3**(4), 150703.

Wikle, C. K. & Berliner, L. M. (2007), 'A bayesian tutorial for data assimilation', *Physica D: Nonlinear Phenomena* **230**(1-2), 1–16.

Xiang, X., Kennedy, R., Madey, G. & Cabaniss, S. (2005), Verification and validation of agent-based scientific simulation models, *in* 'Agent-directed simulation conference', Vol. 47, p. 55.

Zanella, A., Bui, N., Castellani, A., Vangelista, L. & Zorzi, M. (2014), 'Internet of things for smart cities', *IEEE Internet of Things journal* **1**(1), 22–32.

Zhang, Y., Li, X., Liu, X. & Qiao, J. (2011), 'The ca model based on data assimilation', *Journal of Remote Sensing* **15**(3), 475–491.

Zhang, Y., Li, X., Liu, X., Qiao, J. & He, Z. (2013), 'Urban expansion simulation by coupling remote sensing observations and cellular automata', *Journal of Remote Sensing* **17**(4), 872–886.