System optimum during the evacuation of pedestrians from a building
A minor field study

Oskar Blom Göransson
oskarbg@kth.se

Supervisor: Hairong Dong ¹
Examiner: Xiaoming Hu ²

September 2012

¹State Key Laboratory of Rail Traffic Control and Safety, Beijing Jiaotong University, Beijing, China
²Division of Optimization and Systems Theory, Royal Institute of Technology, Stockholm, Sweden
Abstract

During an emergency in a building complex, an effective evacuation is essential to avoid crowd disasters. The evacuation efficiency could be enhanced both by changing the layout of the building, and by changing the route guiding given to the evacuating pedestrians. This thesis considers how to guide the evacuating pedestrians so that the evacuation time is minimised.

In this thesis, a dynamic network model, namely the point queue model, is used to form a linear programming problem whose solution is used to create an evacuation plan. By continuously updating the initial data in this model and solving the problem with this new data, a feedback based control law is derived based on Model Predictive Control.

The control law is tested on a simulation of the social force model for a building with five rooms and one respectively two exits. The result shows that the control law manages to efficiently guide the pedestrians out of the building, taking the varying distribution of pedestrians into account. The control law further manages to handle minor errors in the layout information.

Keywords. Evacuation modelling, pedestrian dynamics, optimal control
Sammanfattning

Under en nödsituation i en byggnad är en effektiv evakuering nödvändig för att katastrofer ska kunna undvikas. Effektiviteten kan förbättras dels genom att förändra byggnadens struktur och dels genom att förändra utrymnningen av människor från byggnaden. Det här examensarbetet rör hur guidningen skulle kunna utformas så att evakueringstiden minimeras.

En dynamisk nätverksmodell (the point queue model) anpassas för att modellera flödet av människor i en byggnad och från den formuleras ett linjärrangeringsproblem vars lösning används för att skapa en evakueringstrategi. Genom att kontinuerligt uppdatera initialvärdena i modellen och lösa problemet på nytt kan en feedbackbaserad kontrol skapas genom Model Predictive Control. Kontrollgen testas på en simulering av ”the social force model” i en byggnad med fem rum och en respektive två utgångar. Resultatet visar att kontrollen ger effektiva evakueringsrutter som tar hänsyn till hur distribueringen av människor varierar med tiden. Vidare klarar den, tack vare feedback, av att hantera vissa felaktigheter i informationen om byggnadens utseende.

Nyckelord. Evakueringsmodellering, modellering av gångare, optimal kontroll
Acknowledgements

I want to thank professor Hairong Dong for giving me the opportunity to work under her supervision at the Jiaotong University in Beijing. For always having time to listen to my problems and, by her wise advices, pushing me in the right directions.

I want to thank my examiner, professor Xiaoming Hu, for giving me this opportunity and for his help and discussions, both during the preparations and during the work with the thesis. He always gave me valuable feedback and encouragement.

I also want to thank all the students working under prof. Dong as well as my other co-workers at Jiaotong University for their invaluable help. They listened to my ideas and came with relevant questions that pointed out the weak points of my argumentation as well as helping me to solve the problems that I encountered. Further, they spent hours and hours helping me with all possible practical problems that I encountered while staying in Beijing. Without this help, I do not no what I would have done.

I also want to thank SIDA for accepting this project as a minor field study and thus making it financially possible for me to conduct this study in Beijing.
## Contents

1 Introduction 1

2 Pedestrian dynamics – models and behaviour 3
   2.1 Fundamental diagram 3
   2.2 The Social Force Model 5
      2.2.1 Modelling pedestrian behaviour using forces 5
      2.2.2 Calibration and parameter estimation 8
   2.3 Cellular Automata models 8
      2.3.1 Discussion 10
   2.4 Vision based methods 11
      2.4.1 Shortest path search 11
      2.4.2 Bearing angle and time-to-collision 13
      2.4.3 Collision predictive force model 13
      2.4.4 Discussion 14
   2.5 Hydrodynamical models 14
   2.6 Generalised kinetic theory model 15
      2.6.1 Discussion 17
   2.7 Optimising layouts using pedestrian simulations 17
      2.7.1 Placements of desks in a classroom 18
      2.7.2 Randomly created obstacles outside door 18
      2.7.3 Usability of optimal layouts 19

3 Theory 20
   3.1 Model predictive control 20
      3.1.1 Mathematical formulation of MPC 21
      3.1.2 Stability of closed-loop control law 23
   3.2 Point queue model – the predictive model 25
      3.2.1 Continuous dynamic network flow models 25
      3.2.2 Governing equations in PQ-model 27
      3.2.3 Discretising the PQ-model 28
      3.2.4 Formulation of a linear programming problem 28

4 Constructing model 30
   4.1 Modelling the social force model 30
      4.1.1 Modelling the wall force 30
      4.1.2 Modelling the social force 31
      4.1.3 Modelling the driving force 32
      4.1.4 Updating position 32
Chapter 1

Introduction

During an emergency in a building complex, an effective evacuation is critical to avoid crowd disasters. It is therefore problematic that humans in emergency situations tend to behave in a way that prolongs the evacuation, for example by following the crowd and thus not using all exits efficiently. The wish to escape and the consequent emerging panic further leads people into both moving faster than normal and pushing when the surrounding crowd slow them down. This might create clogging at doors and other narrow passages where: first, fewer people can pass than under normal conditions; and second, dangerously high pressure from the surrounding crowd might crush people, especially if they fall, causing what is sometimes referred to as stampede. (Helbing et al., 2000; Schadschneider et al., 2008)

In order to prevent crowd disasters, an important first step is to accurately predict the evacuation process. With this achieved, dangers can be predicted and efficient precautions taken, and many models describing pedestrian movement and flow have been developed with this purpose. Initially, the pedestrians were treated macroscopically, as a flow, but with the increase in computer capacity, it has become possible to construct microscopical models where every pedestrian is treated individually. In these models, the combined individual behaviour has reproduced observed collective behaviour, for example lane formation in a bidirectional path and clogging at bottlenecks for high densities (Schadschneider et al., 2008). The most famous such model is perhaps the social force model (Helbing and Molnár, 1995).

There are, however, limitations on what can achieved merely by changing the layout in accordance with these predictions. An important second step is therefore to control the evacuation process by guiding the pedestrians towards safe routes. In practise, this can mean guiding the pedestrians towards exits using emergency exit signs and trained personnel. However, this has two main problems: First, it is impossible to predict where the accident causing the emergency occurs and thus to know which routes that will be safe. Second, it cannot be assumed that all pedestrians will follow the directions given, and even if they did, the consequences cannot be exactly predicted. A static escape route might thus lead to congested and unsafe routes.

The ideal would be to guide the pedestrians to the exits based on updated data on the security situation on different routes and on the current distribution and prediction of future distribution of pedestrians. The route guiding should
thus be dynamic and based on feedback from the actual evacuation. The purpose
of this thesis is therefore to:

(i) Develop a control law for the route choice that ensures an efficient evac-
uation of pedestrians from a building complex. The control law should
include feedback and be based on a prediction of the future behaviour of
the pedestrians.

(ii) Apply the control law to a microscopic pedestrian simulation to evaluate
the result.

Since microscopic models generally are more realistic than macroscopic mod-
els, it would be ideal if a control law could be derived directly from such a model.
However, since microscopic models typically struggle to compute simulations in
real time, this is not a realistic alternative, especially if the solution is to be used
as feedback. Another approach towards modelling the evacuation of a building
complex would be to represent the different escape routes by links and nodes
in a dynamic network traffic flow model. This can provide the possibility to
calculate the optimal evacuation strategy for a complex building as a solution
to a linear programming problem. The dynamic network model might not com-
pete with microscopic models in its ability to correctly predict the evacuation
process and the consequences of different building layouts, but it might be “good
enough” when feedback is applied.

This thesis will start with a literature review, containing an overview of
different pedestrian models, as well as examples of how optimisation has been
applied to pedestrian simulations in order to create efficient layouts from an
evacuation perspective. Since no available pedestrian model is deemed appro-
piate for finding optimal routes, this is followed by a chapter concerning the
theory needed to develop such a model as well as model predictive control the-
ory. After this, how the microscopic simulation that is used for verifying the
control law was done in practise and how control was applied to it will be de-
scribed. Finally, the scenarios used for testing the control law will be discussed
followed by result and a discussion of the result, a direction towards future work
and possible improvements.
Chapter 2

Pedestrian dynamics – models and behaviour

2.1 Fundamental diagram

The fundamental diagram is the relation between the specific flow $J_s$ and pedestrian density $\rho$. The specific flow is the flow of pedestrians per second per meter, and the density is the number of pedestrians per square meter. The flow is of course dependent on the density. At low densities, the speed with which pedestrians can move will only be slightly influenced by other pedestrians and therefore the specific flow will increase with the density. Above a certain threshold, the influence of other pedestrians will be too big and the pedestrian flow will decrease.

A lot of empirical research has been done with the aim to correctly describe this relation and a good overview is given in Schadschneider et al. (2008). There are so far no conclusions of how the relationship should accurately be expressed analytically, mostly because the empirical results are so inconclusive. Results from empirical studies gives $J_{s,max}$ varying between $1.2/\text{m s}$ and $1.8/\text{m s}$, density $\rho_c$ corresponding to this maximum flow varying between $1.75/\text{m}^2$ and $7/\text{m}^2$ and density $\rho_0$ for which the velocity approaches zero varying between $3.8/\text{m}^2$ and $10/\text{m}^2$. These huge variations has been explained as cultural differences; difference between uni- and bidirectional flow; different purpose for walking, like shopping, walking between stations or escaping an emergency; etc.

Of natural reasons a lack of data from actual evacuations and emergencies. This makes it hard to determine realistic values of parameters and realistic fundamental diagram to calibrate models used for predicting the evacuation process quantitatively. One exception is Helbing et al. (2007) where data was actually gathered from a crowd disaster. In the report, video recordings from the Muslim pilgrimage in Mina/Makkah – that resulted in a crowd disaster in the form of a stampede – were analysed and fundamental diagrams extracted. They measured $J_{s,max} \approx 2/\text{m s}$ for $\rho_c$ between $2/\text{m}^2$ and $5/\text{m}^2$ and $\rho_0 \approx 6/\text{m}^2$.

Although the situation differ from that of an evacuation, it is a very rare example of data from an actual emergency and is therefore important to consider in evacuation research.

Two analytical expressions of the fundamental diagram are commonly found
in literature. The most simple expression – sometimes called Greenshield’s model (Kachroo et al., 2008) – models the velocity \( v(\rho) \) as linearly decreasing with the density \( \rho \). This means that:

\[
v^G(\rho) = v_{ffv}(1 - \frac{\rho}{\rho_0})
\]  

(2.1)

where \( v_{ffv} \) is the free flow velocity and with \( \rho_0 \), as above, defined by \( v(\rho_0) = 0 \). From eq. (2.1), the corresponding specific flow is given from the relationship

\[
J_s(\rho) = v(\rho)\rho
\]

A more sophisticated expression is given by Weidmann’s model. The following form is found in Johansson et al. (2008):

\[
v^W(\rho) = v_{ffv}(1 - e^{c_W(\frac{1}{\rho} - \frac{1}{\rho_0})})
\]  

(2.2)

where the following values are used: \( v_{ffv} = 1.34 \text{ m/s}, \rho_0 = 5.4 / \text{m}^2 \) and the fit parameter \( c_W = 1.913 \text{ m}^2 \).

Using the values above, figure 2.1 can be obtained. From the figure it is clear that Weidmann’s model gives a considerably smaller \( J_{s,max} \) than Greenshield’s, and that the corresponding \( \rho_c \) is smaller still. Some characteristics in the velocity dependence in Weidmann’s model definitely makes it more attractive, especially that the velocity does not decrease until a certain density is reached. However, from the empirical studies it is hard to find support for any one of these models. As said before, an overview of empirical research is given in Schadschneider et al. (2008).
2.2 The Social Force Model

In the social force model, pedestrians are modelled as particles that affect each other and are affected by the environment by forces. The model was proposed in Helbing and Molnár (1995) and since then it has been thoroughly tested and further developed.

This section will start with an introduction of how the pedestrian behaviour is represented by forces in the model and the result that was obtained when using different forms of these forces. Following that, some calibration attempts will be given.

2.2.1 Modelling pedestrian behaviour using forces

The forces acting on the pedestrians are usually divided into three categories: the internal driving force $f_i$; the social (or interpersonal) forces $f_{ij}$, including a psychological force and a physical force; and the object force, once again including a psychological force and a physical force. By adding these forces together, the acceleration of a pedestrian $i$ is given by:

$$m_i \frac{dv_i}{dt} = f_i(t) + \sum_{j \neq i} f_{ij}(t) + \sum_W f_{iW}(t) \quad (2.3)$$

**The driving force.** This is the force that accelerates the pedestrian against its target destination. By definition, the force is strong enough to accelerate the pedestrian from the current velocity to the desired velocity within the relaxation time $\tau$. With $e_0^i(t)$ as the desired direction of motion, $v_0^i$ as the desired speed and $v_i(t)$ as the current velocity, the driving force is calculated from (Helbing and Molnár, 1995):

$$f_i(t) = v_0^i e_0^i - v_i(t) \quad (2.4)$$

**The social force.** This force is the way in which the social force model deals with human interaction. It consists of two parts: Firstly a psychological force $f_{ij}^{psych}$. This force models the desire to keep a safe-distance to other pedestrians in order to avoid collisions. It is thus not a real force but an attempt to give an easy representation of the human route choice that in reality is an advanced psychological process. Secondly, it consist of a physical force, $f_{ij}^{phys}$, that models the physical interaction between two pedestrians. Although physical interaction is avoided through the psychological force, it might occur when high number of pedestrians are gathered in small areas. How these two forces are modelled vary somewhat in the literature and the representation has been developed and calibrated over the years to better reproduce observed data.

One way to model the social force that repels pedestrian $i$ from pedestrian $j$ is given by (Helbing et al., 2000):

$$f_{ij}^{psych} = Ae^{r_{ij} - d_{ij}} n_{ij} \quad (2.5)$$

where $A$ and $B$ are coefficients that are chosen so the resulting behaviour is realistic; $r_{ij} = r_i + r_j$ is the sum of the radius of pedestrian $i$, $r_i$, and pedestrian $j$, $r_j$; $d_{ij}$ is the distance between the centre of pedestrian $i$ and $j$; and $n_{ij}$ is
CHAPTER 2. PEDESTRIAN DYNAMICS – MODELS AND BEHAVIOUR

\[ \mathbf{v}_j \]

\[ \mathbf{n}_{ij} \]

\[ \mathbf{p}_i \]

\[ \mathbf{f}_{ij} \]

\[ \mathbf{d}_{ij} \]

\[ \psi_{ij} \]

\[ \lambda \]

\[ \Theta(\psi_{ij}) = \left( \lambda + (1 - \lambda) \frac{1 + \cos(\psi_{ij})}{2} \right) \]

(2.6)

\[ f_{\text{psych, an}} = \Theta(\psi_{ij}) Ae^{\frac{\mathbf{e}_i^0 - \mathbf{e}_j^0}{\psi}} \mathbf{n}_{ij} \]

(2.7)

The force in eq. (2.7) only models the psychological tendency among pedestrians to avoid each other. This is modelled by adding further force terms: firstly a normal force representing the force between the pedestrians when they are pushed towards each other; and secondly a tangential force representing the...
friction occurring if they are moving alongside (Helbing et al., 2000). These physical forces are only nonzero if the pedestrians are in contact. To take this into account, the indication function \( g(r_{ij} - d_{ij}) \) is added and defined as (Helbing et al., 2000):

\[
g(r_{ij} - d_{ij}) = \begin{cases} 
    r_{ij} - d_{ij} & \text{if } r_{ij} - d_{ij} \geq 0, \\
    0 & \text{if } r_{ij} - d_{ij} < 0,
\end{cases}
\] (2.8)

where \( r_{ij} - d_{ij} \geq 0 \) indicates that pedestrian \( i \) and \( j \) are in contact since the distance \( d_{ij} \) between their centre positions is less than their combined radius \( r_{ij} \).

The physical force between two pedestrians is further dependent on the difference in velocity between the two pedestrians. If they are moving with the same velocity, the frictional force should be zero. The difference in tangential velocity is therefore called \( \Delta v_{ji} = (v_j - v_i) t_{ij} \). In Helbing et al. (2000), the physical force exerted on pedestrian \( i \) as a consequence of the physical contact with pedestrian \( j \) is given by

\[
f_{\text{phys}}^{ij} = kg(r_{ij} - d_{ij}) n_{ij} + \kappa g(r_{ij} - d_{ij}) \Delta v_{ji} \mathbf{t}_{ij} \] (2.9)

where \( \mathbf{t}_{ij} = (-n_{ij}^2, n_{ij}^1) \) is the tangential direction, \( k = 1.2 \cdot 10^5 \text{ kg/s}^2 \) and \( \kappa = 2.4 \cdot 10^5 \text{ kg/ms} \). Combining eq. (2.9) and (2.7), gives the force on pedestrian \( i \) from pedestrian \( j \) as:

\[
f_{ij} = \Theta(\phi_{ij}) A e^{-d_{ij}} n_{ij} + kg(r_{ij} - d_{ij}) n_{ij} + \kappa g(r_{ij} - d_{ij}) \Delta v_{ji} \mathbf{t}_{ij}. \] (2.10)

As mentioned before, many different formulations of the social force have been proposed. For example, it was not able to reproduce the later discovered crowd phenomenon crowd turbulence (Yu and Johansson, 2007). Therefore, the social interaction function in eq. (2.10) was modified to:

\[
f_{ij} = F \Theta(\phi_{ij}) A e^{-d_{ij}} n_{ij} + kg(r_{ij} - d_{ij}) n_{ij} + \kappa g(r_{ij} - d_{ij}) \Delta v_{ji} \mathbf{t}_{ij}. \] (2.11)

where \( D_0 \) and \( D_1 \) are new constants; and the tangential and normal physical force are removed. An alternative way of handling the anisotropy of the social force was proposed already in Helbing and Molnár (1995). This is called an elliptical specification of the social force model and the social interaction equation in this specification is much more complex. This complexity makes it hard to calibrate and more computational demanding and it was therefore overlooked for a long time, but as the limitations of other formulations become clearer, the elliptical specification gets more attention. One such limitation of the social force function in eq. (2.10), discussed in Ondřej et al. (2010) and Karamouzas et al. (2009), is that a group of pedestrians moving towards another group will not be able to navigate through the other group in any practical way. Another limitation is that it is hard to get both good individual trajectories and fundamental diagram at the same time, something that can be done better with an elliptical specification (Johansson et al., 2008).

**The object force.** The effect of objects on the route choice of pedestrians is modelled similarly to the effect of other pedestrians. One force term models the psychological tendency of pedestrians to avoid contact with and keep a safe
CHAPTER 2. PEDESTRIAN DYNAMICS – MODELS AND BEHAVIOUR

distance to objects and one term models the physical forces that affect the pedestrian if in contact with an object. Using the same approach that gave to the social interaction force in eq. (2.10) the total force $f_{iW}$ can be derived to (Helbing et al., 2000):

$$f_{iW} = A_i e^{-\frac{x - d_{iW}}{d_i}} n_{iW} + kg(r_i - d_{iW}) n_{iW} - kg(r_i - d_{iW})(v_i \cdot t_{iW}) t_{iW}. \quad (2.12)$$

2.2.2 Calibration and parameter estimation

Numerous attempts have been made to calibrate the social force model to better reproduce observed phenomena, both qualitatively and quantitatively. Since different representations of the forces have been used, it is hard to give a comparison of the result, but an overview of some calibration attempts are given below to give an idea of the methods that have been used to estimate the parameters used.

By asking students to walk back and forward through an empty corridor and measuring their trajectories, and calibrating the acceleration behaviour determined by the driving force to the recorded data, Moussaid et al. (2009) estimated the relaxation time to $\tau = 0.54 \pm 0.05$ s. The value $\tau = 0.50$ s, initially proposed by Helbing and Molnár (1995), is thus included in the standard deviation.

The desired velocity $v^0_i$ is of course dependent upon both the situation and the individual. For example, in an emergency situation it is likely that pedestrians want to move faster than under normal conditions. Different values of this velocity, usually labeled $v^0_i$ for pedestrian $i$, have been proposed. (Helbing et al., 2000) suggests that a desired velocity of $v^0_i = 1.5$ m/s or higher should be used to model emergency conditions. In their experiments (Moussaid et al., 2009) measured a medium velocity of $v^0_i = 1.29 \pm 0.19$ m/s during normal conditions.

In Helbing and Johansson (2010), an attempt to calibrate the anisotropy constant $\lambda$ from eq. (2.6) is made using video recordings of pedestrian trajectories giving $\lambda = 0.1$.

In Helbing et al. (2000), the constants of the interpersonal force equation in eq. (2.5) are chosen to $A = 2 \cdot 10^3$ N and $B = 0.03$ m since this reproduced data on safe distances kept and bottleneck flows. In Johansson et al. (2008), optimisation is used to calibrate the constants governing the social force model with anisotropy to trajectories of individual pedestrians moving in a crowd. A group of combinations of $A$ and $B$ was discovered to give equally good result. One possible solution was $A = 0.2$ N and $B = 1$ m. It is notable how big the difference was between these two calibrations, and this highlights the difficulty of calibration the social force model.

2.3 Cellular Automata models

Cellular Automata (CA) models represent another type of microscopic pedestrian models. In CA models, space is discretised into a grid where, typically, each cell in the grid can be occupied by at most one pedestrian, as in figure 2.3. The time is also discretised into time steps where a time step usually represent the time it takes for a pedestrian to move to an adjacent cell with a “normal” velocity. The position of all pedestrians are updated simultaneously,
Figure 2.3: An example of the possible movements (left) and corresponding transition probabilities (right) for a CA model.

and thus there might occur conflicts when pedestrians try to move to the same cell. Below, the construction of a CA model for pedestrian behaviour will be described.

**Space discretisation.** The most common way to discretise the space is to let each cell represent $40 \times 40$ cm, since this is regarded to be the space a human occupies in a dense crowd (Burstedde et al., 2001). With this discretisation, each pedestrian occupies one cell in the grid. For an example of the discretisation, see figure 2.3.

**Time discretisation.** The time scale of the model is based on a presumed reaction time of pedestrians. The idea is that it takes $t_{\text{reac}}$ for pedestrians to react on actions of other pedestrians. Therefore, if two pedestrians are attempting to enter the same cell, at least one of the pedestrians will neither be able to move there nor be able to come up with a new moving direction until the next time step. The average velocity of pedestrians can roughly be estimated to $v \approx 1.3 \text{ m/s}$ and since a pedestrian only can move to an adjacent cell in each time step, this gives $t_{\text{reac}} = 0.3 \text{ s}$, which is close to human reaction time. Observe that if the pedestrians also are able to move diagonally, as indicated in figure 2.3, the speed will differ depending on the moving direction.

**Pedestrian movement.** The movement of the pedestrians in the CA models is determined by the transition matrix, again see figure 2.3. The values of the transition matrix determines the probability that the pedestrian will attempt to move to respectively cell. A common way of modelling the transition matrix is by introducing a floor field (Burstedde et al., 2001; Kirchner and Schadschneider, 2002). The strength of the floor field is often given by a combination of a static and a dynamic floor field. The static floor field is calculated in the beginning of the simulation and its value at a cell is dependent of the cells distance to attractive regions such as exits in an evacuation simulation. The dynamic floor field changes, as the name implies, with time. It is added to take the current and previous distribution of pedestrians into account.

How the pedestrians are allowed to move and how the floor field value is calculated varies between different models. Some models only allow movements
in four directions (up, down, left or right) whereas other allow for movement to all eight neighbouring cells. Further, some models are stochastic in their nature. In these models, the floor field value at a cell relates to the probability that a pedestrian in an adjacent cell will try to move there, and stochastic variables are used to decide which cell pedestrians chose. In non-stochastic models, pedestrians choose the adjacent cell with the lowest (or highest, depending on model) value of the floor field.

One way to model the static floor field $S_{ij}$ for a cell with coordinates $(i,j)$ is by the distance to the closest exit. The cells $\{(i_{T1},j_{T1}),\ldots,(i_{Tk},j_{Tk})\}$ represent $k$ different exits. The static floor field $S_{ij}$ can then be defined by the metric (Kirchner and Schadschneider, 2002; Xu et al., 2011):

$$S_{ij} = \min_{i_{Ts},j_{Ts}} \max_{i_l,j_l} \left\{ \sqrt{(i_{Ts} - i_l)^2 + (j_{Ts} - j_l)^2} - \sqrt{(i_{Ts} - i)^2 + (j_{Ts} - j)^2} \right\}.$$  \hfill (2.13)

In the equation, $\max_{i_l,j_l} \left\{ \sqrt{(i_{Ts} - i_l)^2 + (j_{Ts} - j_l)^2} \right\}$ normalise the distance. It is the furthest away any cell is from target $s$. This means that the static floor field reach a maximum at the exit cells. This representation of the static floor field is “exact” in the sense that its value at a cell corresponds to the distance from the cells position in the real world to the target destination in the real world. Note that there is a discrepancy between the real world length and the length a pedestrian needs to traverse since the movement of pedestrians are highly restricted. If there are obstacles in a room, this static floor field must be changed to take them into account, something that can be achieved with the introduction of visibility arcs Kretz et al. (2010).

Another way to create the static floor field is by using so called flood fill methods, for example based on the Manhattan metric. In these methods, the field of the grid is calculated by selecting starting cells, and then sequentially moving to adjacent cells and adding up the distance as one moves along. These methods are very computationally effective but are not exact in the sense that the metric defined by eq. (2.13) is exact. For a survey of different methods of creating the static field and a comparison of their errors and computational demand, see Kretz et al. (2010).

In (Burstedde et al., 2001; Kirchner and Schadschneider, 2002), the dynamic floor field is formed by letting each pedestrian leave an attractive virtual trace particle, called a boson, in the cells they occupy. In every time step, a stochastic variable determines whether or not a boson: remains at it’s current position; decays, i.e. is removed completely; or diffuses, i.e. moves to an adjacent cell. An advantage with this method is that the path choice of the pedestrians are determined by a global field that only needs to be calculated once per time step.

### 2.3.1 Discussion

This discretisation put some limitations on the use of CA models. First of all, the speed with which the pedestrians are able to move is generally constrained to one cell per time unit. To change this without changing the space discretisation, either the time scale needs to be doubled or the pedestrians will move with twice the speed. Secondly, the size of spaces, doors and obstacles will be limited to even multiples of 40 cm, which definitely poses a limitation if the model is to be used to compare different layout options against each other. It is
worth mentioning that attempts have been made to introduce a finer grid size, for example in Kirchner et al. (2004), where a discretisation is proposed in which each pedestrian occupies four cells. This discretisation comes with higher computational demands and with a more complex solution to how to update the position of pedestrians.

Further, it is hard to validate the rules determining the movement of pedestrians in the models, other than by the fact that it gives somewhat realistic results. Even though the virtual trace dynamic floor field method has produced many observed crowd phenomena—such as realistically looking clogging and lane formation (Schadschneider et al., 2008)—it is questionable whether a virtual trace is a realistic way to model the movements of pedestrians. Further, since the social force model is able to reproduce these phenomena as well, in a more realistically looking way, the main reason for considering usage of a CA-model would be the relative computational complexity of the social force model. Since it is possible to run simulations of thousands of pedestrians in real time even with the social force model, the main reason for using a CA-model would be during evacuation of really large areas or for optimisation. But, as discussed above, it is questionable how well suited the CA-model is for proposing good building layouts, which has been the main application area for optimisation on pedestrian models.

2.4 Vision based methods

The social force model and CA models try to find ways of simplifying how humans find their way through a crowd, and even though simplifications are made, good enough results can be obtained at a relative low computational cost. As discussed, experiments and calibrations have shown that these models are able to reproduce a number of crowd phenomena as well as quantitatively reproduce flow data. But it is impossible to escape the fact that these models greatly simplify the reality, and that there is a limit for how well they can perform. If one was able to capture the real essence of pedestrian navigation, that would promise more realistic models. In reality, the main source of information used by pedestrians is their vision. This information is used to predict the future position of other moving objects and based on this find smooth paths that avoids contact with the objects. Below, three different models starting from this perspective is described.

2.4.1 Shortest path search

In Moussaïd et al. (2011), repulsive psychological forces as those in eq. (2.7) and eq. (2.12) in the social force model is removed in favour of a scan of the field of view for the shortest path to the destination. Two behavioural heuristics are proposed to govern the navigation of a pedestrian. These heuristics govern the choice of desired walking direction \( e_i^0(t) \) and desired walking velocity \( v_i^0 \), where the later will be modelled by the dynamic variable \( v_{des}^0(t) \). Thus, it is a driving force \( f_i \) as in (2.5) that guides the pedestrian towards a free path, rather than a repulsive collision avoidance force.

The first heuristic states that: a pedestrian will move in the direction that allows for the shortest path to the target destination, including objects and the fu-
ture position of other pedestrians. This is used to determine the desired walking direction \( \mathbf{v}^0_i(t) \). The heuristic is motivated by the observation that pedestrians search for an unobstructed walking direction, while at the same time trying to avoid to big deviations from the direct path to the target destination. In practise, this is modelled in the following way: First, the field of view is defined as the area within the horizon distance \( d_{max} \) of the pedestrian and between the angels \( \alpha \in [-\phi, \phi] \), for some reasonable angle \( \phi \) and with \( 0^\circ \) as the current walking direction. Second, the distance to the first collision in direction \( \alpha \) is measured by the function \( f(\alpha) \) to the longest distance the pedestrian could walk with the desired velocity \( \mathbf{v}^0_i \) in that direction until a collision would occur, when the future position of other pedestrians are taken into account. If no collision is predicted to occur within the field of view, \( f(\alpha) = d_{max} \).

Suppose that the target destination in the field of view for the pedestrian is at angle \( \alpha_0 \). Then the first heuristic is modelled by letting the pedestrian choose the direction \( \alpha_{des}(t) \) which minimises the distance function \( d(\alpha) \), i.e. :

\[
\alpha_{des}(t) = \arg \min_{\alpha} d(\alpha) \quad (2.14)
\]

where

\[
d(\alpha) = d_{max}^2 + f(\alpha)^2 - 2d_{max}f(\alpha)\cos(\alpha_0 - \alpha) \quad (2.15)
\]

can be obtained using the law of cosine as the distance left to walk after walking the distance \( f(\alpha) \) in the direction \( \alpha \).

The second heuristic states that: a pedestrian will maintain a safe-distance to other pedestrians and objects such that the pedestrian will be able to stop before they are predicted to collide. Suppose that the distance to first collision is \( d_h \) and that the reaction time, as in the social force model, is \( \tau \). Then this second heuristic can be modelled by letting the desired walking speed be given by \( v_{des}(t) = \min(v_i^0, d_h/\tau) \), where \( v_i^0 \) is the desired walking speed if no obstacle exists.

Physical interaction with other pedestrians and obstacles are treated much in the same way as in the social force model. The physical force on pedestrian \( i \) as a consequence of physical contact with pedestrian \( j \) is given by:

\[
f_{ij}^{phys} = kg(r_{ij} + d_{ij})n_{ij} \quad (2.16)
\]

where \( g, d_{ij}, r_{ij} \) and \( n_{ij} \) is defined as in section 2.2. The physical force between pedestrian \( i \) and object \( w \) is given by

\[
f_{w}^{obj} = kg(r_i + d_{iw})n_{ij} \quad (2.17)
\]

where \( r_i \) is defined as before.

The following result were reported when using this model: First, on a local level, the model predicts the collision avoiding behaviour of pedestrians walking through a corridor with an obstacle in the middle well, when compared to video trajectories of real pedestrians. Since no parameters influence this behaviour, this is a validation rather than a calibration of the model. Second, on a crowd level, the model is able to reproduce lane formation in a bidirectional street; crowd turbulence and stop and go waves at higher densities; and the resulting fundamental diagram realistically.
2.4.2 Bearing angle and time-to-collision

Another collision prediction approach is given Ondřej et al. (2010). The method is based on social psychological studies that suggest that pedestrians answer two questions when interacting with moving and static obstacles. The first question is: *will a collision occur?* This question is answered by observing how the visual or **bearing angle** $\alpha$ between the pedestrian and the object changes with time, i.e. by studying $\dot{\alpha}$. The second question is: *when will a collision occur?* This is answered by studying the rate of growth of the obstacles, i.e. how fast the object is growing in the vision field. If the rate of growth is positive, a collision will occur. If the rate of growth is high, it means that the collision is close in time. In this way, pedestrians are able to approximate the time to collision, $ttc$. If $ttc$ is negative, it means that the object is moving away from the pedestrian, so no collision avoiding action is needed. On the other hand, if $ttc$ is positive and $\dot{\alpha}$ is close to zero, a collision is soon unavoidable and the pedestrian must either change moving direction or change speed. When a collision is predicted to occur, a set of rules is applied to govern the action of the pedestrians. These rules will not be covered here.

This model is developed to create realistically looking simulations rather than for evacuation prediction simulations. Verifications of the model has therefore focused on investigation how realistic it looks, rather than if it is able to reproduce crowd phenomena, and the behaviour looks realistics. For example, when two groups of pedestrians meet, they are able to find smooth ways through each other, something that groups in the social force model was unable to do.

2.4.3 Collision predictive force model

To avoid the extra computational cost that usually is demanded by vision based methods – since they scan the view field and determine a path choice according to this – Karamouzas et al. (2009) uses an approach based on the social force model. In the social force model, pedestrians are affected by a force that is dependent upon the current position of other pedestrians. In this collision predictive force model, the future position of pedestrians is approximated using linear interpolation. If another pedestrian is approximated to enter the **personal space** within the anticipation time $t_\alpha$, a social force is exerted on the pedestrian based on their relative positions at the moment of entry. The concept of a personal space is added to the model via a function $B$, defined in a similar way to the function $g$ in (2.8).

Comparing with the social force model, the movement of pedestrians in a crowd is smoother and pedestrians adopt their speed to avoid collision earlier using this collision predictive force model. As a consequence of this, the pedestrians move with a higher medium speed. The model is developed to be used for animation of human movement, and thus realistic looking movement is the main interest. Therefore, neither emergence of crowd phenomena nor reproduction of fundamental diagram are studied extensively in Karamouzas et al. (2009). The only tested phenomena is lane formation in a bidirectional flow, which the model managed to produce.

The computational demands of the model was reported to be such that 1,000 pedestrians could be modelled in real time using a normal PC.
2.4.4 Discussion

The above presented vision based methods definitely propose interesting ideas for how pedestrian models could be developed. The result presented indicates that these models are more realistic than the social force model and further points to some problems with the social force model. However, two problems with these models are: the lack of validation of the models, except for the shortest path search model; and the lack of data concerning the computational demand, except for the collision predictive force model. Considering this, it is hard to give a fair comparison.

The vision based models are partly introduced to give a more realistic representation of the pedestrian navigation process. However, only in Ondřej et al. (2010) does the reading of social psychological research affect the modelling.

2.5 Hydrodynamical models

All pedestrian models considered so far has modelled the behaviour of every pedestrian individually. This has the advantage of allowing for detailed analysis on individual level but the disadvantage of long computation time. If less precision is needed, one might consider macroscopic models. These are either based on generalised hydrodynamical equations, considered in this section; or based on the equations in kinetic theory of gases, considered in the next section. The macroscopic models provide a way of overlooking the details of while still providing a realistic fundamental diagram Kachroo et al. (2008). Further, it is easier to develop and apply feedback control laws on macroscopic models, since control applied to microscopic models demands the control of every single pedestrian individually Kachroo et al. (2008). During evacuation, pedestrians are assumed to follow the group and therefore move in crowds. Therefore, macroscopic models could be well suited to model emergency evacuation.

An overview of current macroscopic models is given in Bellomo and Dogbe (2011) and an example of how to apply control theory to hydrodynamic models to optimise the evacuation of pedestrians from a single room is given in Kachroo et al. (2008). An introduction to the mathematical models governing these methods will be given below.

Hydrodynamical models are based on mass and momentum conversation equations, two coupled partial differential equations that determine the dynamics of the system. These models were originally developed to describe traffic flow and since traffic flow in many situations can be described well using a one dimensional model, most models proposed are initially one dimensional. Since pedestrians move in a two dimensional space, a good pedestrian model should be two dimensional. A typical way of choosing the governing equations for the flow of pedestrians in two dimensions is given by (Bellomo and Dogbe, 2011):

\[
\begin{align*}
\partial_t \rho(t, x) + \partial_x \left( \rho(t, x) v(t, x) \right) &= 0 \\
\partial_t v(t, x) + (v(t, x) \cdot \nabla_x) v(t, x) &= A[\rho, v] \tag{2.18}
\end{align*}
\]

where the first equation is the mass conservation equation and the second is the momentum conversation equation. Here \(\rho(t, x)\) is the density and \(v(t, x)\) the velocity of pedestrians at point \(x\) and time \(t\); and \(A[\rho, v]\) models the acceleration of the system. What differs models is mainly the choice of \(A[\rho, v]\).
CHAPTER 2. PEDESTRIAN DYNAMICS – MODELS AND BEHAVIOUR

To model the desire of pedestrians to move toward exits, a desired velocity vector is created towards which the velocity of the flow is adapted. The velocity should further be dependent on the distribution of pedestrians, and the velocity adaption as a consequence of this is modelled by another vector field. A simple way of modelling this in practise is by letting \( \mathbf{A} \) consist of two parts: the frictional acceleration \( \mathbf{A}_F \), proportional to the difference between the mean velocity of a the crowd and the actual velocity of a pedestrian; and the acceleration \( \mathbf{A}_P \), proportional to the gradient of the pedestrian density. This gives:

\[
\mathbf{A}_F = c_F (v_c(\rho) - v) \tag{2.19}
\]

where \( \mathbf{e}(t, x) \) is the unit vector in the desired direction, and the acceleration

\[
\mathbf{A}_P = -c_p \nabla_s \rho. \tag{2.20}
\]

By letting these two forces act in the direction of the target destination, the following is obtained:

\[
\mathbf{A} = (\mathbf{A}_P + \mathbf{A}_F) \mathbf{e}(t, x). \tag{2.21}
\]

The above described model is a very simple example of a hydrodynamic model of pedestrian behaviour, but it serves as an example for how different models are developed. For a more detailed review of different methods and the result obtained by using these, the reader is referred to the references and literature therein (Bellomo and Dogbe, 2011; Kachroo et al., 2008).

2.6 Generalised kinetic theory model

In Ch. 6-7 of Bellomo and Dogbe (2011), two different generalised kinetic theory approaches for modelling pedestrian dynamics are proposed. A brief outline of these approaches for the one dimensional case will be given below, but for a more complete description and for a two dimensional formulation, readers are once again referred to references and literature therein. It should also be noted that the article focus on vehicular traffic, and only gives an introduction to how these models can be extended to cover pedestrian dynamics. The first approach presented uses a continuous distribution function and assumes homogeneous behaviour to model the pedestrians. In reality, there are not enough pedestrians in a crowd or drivers on a road for this to be realistic. Therefore, a second granular flow approach is also presented.

Kinetic theory were originally developed by Boltzmann and Maxwell, among others, to describe the distribution of particles that are moving freely in space except for collisions with other particles (Perthame, 2004). Partial differential equations developed to describe the evolution of the distribution density function \( f(t, x, v) \). In its most simplest form, the kinetic theory is based on the transportation equation:

\[
\partial_t f(t, x, v) + v \cdot \partial_x f(t, x, v) = 0
\]

describing the evolution of a particle in free space, much in the same way as the mass conservation equation, eq. (2.18), in hydrodynamics. In the Boltzmann equation, a collision operator \( Q[f] \) is added to the function, giving

\[
\partial_t f(t, x, v) + v \cdot \partial_x f(t, x, v) = Q[f](t, x, v)
\]
where \( Q[f](t, x, v) \) is an integral function describing the microscopic physics of collisions. It can be interpreted as the difference between the inflow and outflow of particles into the phase space.

One way to choose \( Q \) for vehicle flow models is:

\[
Q[f](t, x, v) = J_r[f](t, x, v) + J_i[f](t, x, v)
\]

following the same idea that led to the derivation of eq. (2.21) in the hydrodynamic case. Here, \( J_r[f] \) accounts for speed change according to some specified program of desired velocities, while \( J_i[f] \) accounts for speed change as a consequence of local interactions. No description of a pedestrian dynamics model based on this approach is given in (Bellomo and Dogbe, 2011), but it is stated that in a pedestrian model, \( J_r[f] \) would model the desire of pedestrians to move against a specific target while \( J_i[f] \) would model the change in velocity of pedestrians as a consequence of other pedestrians in the vicinity.

The granular models are motivated by the observation that vehicles and pedestrians tend to move in clusters of the same velocity, and thus the state space of possible velocities should be discrete. The state space discretisation can either be static or dependent on the mass density function. Bellomo and Dogbe (2011)

Discretising the velocity gives:

\[
f(t, x, v) = \sum_{i=1}^{n} f_i(t, x) \delta(v - v_i)
\]

where \( t \) is the time; \( x \) is the position; \( v \) is the speed; \( n \) is the number of discrete velocities; \( f_i \) is the distribution function for the different velocities at \( x \) during time \( t \); and \( \delta \) is Dirac’s delta function. Thus the local mass density is given by:

\[
\rho(t, x) = \int_{\Omega} f(t, x, v) \, dv = \sum_{i=1}^{n} f_i(t, x). \tag{2.23}
\]

As in the continuous case, the model consist of partial differential equations describing the change in the distribution functions \( f_i \) according to:

\[
\partial_t f_i(t, x) + v_i \cdot \partial_x f_i(t, x) = J_i[f; \alpha](t, x) \tag{2.24}
\]

where \( f = \{ f_i \}_{i=1}^{n} \); \( J_i \) is a function representing the interaction between particles and the change between states; and \( \alpha \) is a parameter describing the conditions of the road and skillfulness of the drivers. One way to choose \( J_i \) is:

\[
J_i[f; \alpha](t, x) = \sum_{h=1}^{n} \sum_{k=1}^{n} \int_{D_{xy}} \eta[f](t, y) A_{hk}^i[f; \alpha](t, y) f_h(t, y) f_k(t, y) \omega(x, y) \, dy
\]

\[
- f_i(t, x) \sum_{h=1}^{n} \int_{D_{xy}} \eta[f](t, y) f_h(t, y) \omega(x, y) \, dy.
\]

where the first term is the total number of vehicles entering state \( x \) at time \( t \), and the second term is the total number of vehicles leaving the state \( x \) at time \( t \). Further, \( \eta[f] \sim \frac{1}{1+\rho} \) is the rate of interaction; \( \omega(x, y) \) is a weight function; and \( A_{hk}^i[f; \alpha] \) defines a table of games. The table of games is a set of rules
determining probability for a vehicle with speed \( v_h \) to change speed while interacting with a vehicle of speed \( v_k \). The table of games is determined by the mass density function \( \rho \) and the parameter \( \alpha \).

The equations describing the vehicular traffic is already very complex and the 2-dimensional nature of pedestrian dynamics makes for even more complex equations.

### 2.6.1 Discussion

With correct calibration towards measured fundamental diagrams, the macroscopic models could be able to realistically predict the time needed to evacuate pedestrians. Being less computationally demanding, they thus seem to provide an interesting alternative to microscopic models. However, a big problem with the models is how to formulate the boundary conditions at doors and intersections, making it hard to construct computationally effective models for complex scenarios. For example, in Kachroo et al. (2008), only single rooms are considered. Since the layouts of buildings generally are more complex than this, it is probable that the macroscopic models loses much of their simplicity when applied to most interesting problems.

The generalised kinetic models is definitely an interesting combination of microscopic and macroscopic models, but they seem to be very complex and once again it is questionable how much computational costs one would be able to cut when implementing the models on a larger scale.

The main motivation for using macroscopic models, according to Kachroo et al. (2008), is that they are well suited for the development and implementation of control laws. It is indeed true that it is hard both to apply control theory on the path choice of every single pedestrian and to predict the outcome of a control law directly from a microsimulation; on the other hand, if a control law is to complex to apply to a microsimulation, it is definitely to complex to apply on reality. The construction of a control should be based on what one might actually control in reality, and then tested on a simulation to validate its effects.

As mentioned, a number of different implementations of control theory to hydrodynamic pedestrian models are given in Kachroo et al. (2008). However, only control of the velocity of pedestrians on a single link, i.e. in a single corridor, is considered. How one hope to control the velocity of panicking pedestrians in a real life scenario is not discussed. It is further hard to see how the methods applied by Kachroo et al. (2008) could be used to determine optimal routing strategies in building complex.

To conclude, the macroscopic models presented above are simplifications of the microscopic models, but seems to be to complex to be used to derive optimal route guiding.

### 2.7 Optimising layouts using pedestrian simulations

One of the main implementation of pedestrian models has always been to to determine if a building layout can be considered safe. If it is not deemed safe, the size of exits must be increased. With the introduction of microscopic models, it is possible to analyse the layouts in more details and see where an obstacle
might create clogging during an evacuation. A tempting possibility is therefore to use microscopic models to automatically create optimal building layout.

To do this, one must first discuss what to consider “optimal”, which is simply the minimum of some fitness function. The fitness function is a measure of the quality of the solution. During an evacuation, the objective is to minimise the damage to people. To model the injuries suffered by pedestrians in a realistic manner is not easy, and therefore one might hope to find and optimise other aspects of the evacuation process that can be assumed to influence the injuries. Some different parameters one might consider is: total evacuation time; medium or median evacuation time; safety of different routes, by introducing a weight on routes where the risk is higher; and congestion at bottlenecks, since this both prolong evacuation and induce panic and stampede. These parameters can be extracted from the model and given different weights in a fitness function.

Below, two examples of how optimisation has been used and corresponding choice if fitness function will be discussed.

2.7.1 Placements of desks in a classroom

In Hassan and Tucker (2011), the effects of different placement of desks in a classroom is investigated using heuristic optimisation applied to a CA-model. In their model, half of the pedestrians initially occupying the classroom are moving towards a door located in the left side of the room whereas half of the pedestrian are moving towards a door located in the right side. At each time step, the number of pedestrians that are able to move in their desired direction is counted. From this, a fitness function is generated that promotes movements towards the exits and punishes passivity or movements away from the exit.

Two different heuristic optimisation algorithms, hill climbing and simulated annealing, are used in the article. These two algorithms are chosen since the solution space they search is relatively small. The basic idea behind heuristic optimisation is to search through the solution space for solutions with good fitness by sequentially changing or combining the obtained solutions according to a predefined algorithm. The solution space is searched by randomly moving one desk from the old solutions and compare the fitness for the obtained layouts. Using the algorithm, the evacuation time in the simulation can be decreased significantly. On the other hand, the placement of desks is not very practical and it is therefore questionable how useful the result is for a real life applications. Further, since the discretisation of the space in the CA-model puts limitations on the placement and size of objects, it is not certain that the solutions obtained would be equally good with the real size of objects.

2.7.2 Randomly created obstacles outside door

In Johansson and Helbing (2007), the formation of obstacles in front of exits are investigated using heuristic optimisation to a social force model simulation. Compared to the CA-model based method used in Hassan and Tucker (2011), this method allows for freer placement of obstacles. The initial solutions are created by randomly creating a boolean grid in a discretised space where each space unit either is occupied by an obstacle or not. The probability that a grid point will be occupied by a obstacle is weighted to favour the creation of smooth layouts. After the placement of a predetermined number of obstacles a
smoothing algorithm is applied to create nice looking shapes and avoid islands of small objects in the middle of the room.

The fitness function in Johansson and Helbing (2007) is a function of the outflow rate of pedestrians. Four solutions are created at random and evaluated. The best two are kept and the worst two are changed at random to generate new solutions, that again are evaluated. The presented result suggest that one might considerably increase the outflow rate of pedestrians by including obstacles near an exit where clogging otherwise would occur.

2.7.3 Usability of optimal layouts

It is definitely desirable to use a layouts that are optimal from a safety perspective when designing buildings. However, the layouts also need to be functional during normal use and can therefore not take the very unconventional forms that the above described algorithms produce. Further it is questionable whether the above models are exact enough to produce layouts that would be effective for real pedestrians.

Rather than following the resulting layouts exactly, one could use them as a basis for analysing what kind of objects that smoothens the flow, and it should be noted that this is also done by Johansson and Helbing (2007). To test these structures, one could do empirical experiments under normal conditions, and if these turned out good one could consider using them in reality.
Chapter 3

Theory

In the following chapter, the principles behind the control strategy used for deriving optimal route guiding will be derived. The three main parts of this strategy are: First, a simulation of the social force model. This is the plant in figure 3.1 and should be interpreted as an artificial reality used for evaluating the control law. Second, a dynamic network model, namely the point queue (PQ) model, that simplifies the pedestrian dynamics to link dynamics in a network. Using this model, a linear programming problem can be formulated that determines how pedestrians should be guided. This is the predictive model in figure 3.1, since the model predicts the behaviours of the reality. If this was applied directly to the simulation, it would be open-loop control law without feedback. Therefore, and third, Model Predictive Control (MPC) is used to link the PQ-model with the simulation. Model predictive control is the entire loop in figure 3.1.

The outline of this chapter is the opposite to the above description, starting with the theory behind MPC. Following this, the PQ-model will be described and a linear programming (LP) problem will be formulated based on it. The social force model has been covered in section 2.2. How the network in the PQ-model will be adapted to model an evacuation of pedestrians from a building will be discussed in section 4.2.1; how parameters used in the model are estimated will be discussed in section 4.2.2; and how the cost function is chosen is discussed in 4.2.3.

3.1 Model predictive control

In Model Predictive Control (MPC), the current state of a plant is used as input in a model that predicts the dynamics of the plant. Using this predictive model, the evolution of the plant is predicted until the, typically finite, time horizon $T_h$ and an optimisation algorithm is used to determine an optimal open-loop control law from the predictive model until the time $T_c \leq T_h$. This open-loop control is used as input for the plant until a new measurement is able to give an updated state of the plant. When an updated state is available, a new control law is created based on this and again applied to the plant. The basic MPC control loop, and what respectively part represents in this report, is depicted in figure 3.1. To summarise, MPC can be explained by the following four steps.
CHAPTER 3. THEORY

2. Predictive Model
PQ-model predicts plant
\[ \Rightarrow \]
LP-problem
\[ \Rightarrow \]
Open-loop control law: \( u = \{ u(k), \ldots, u(k+T-1) \} \)

3. Plant
Social force model simulation.
\( u(k) \Rightarrow \) desired doors.

1. State Estimator
Recalculate the state of the plant to fit the simplified predictive model.

\( \hat{x} \)

Figure 3.1: Basic control loop in Model Predictive Control.

(Findeisen and Allgöwer, 2002):

1. Measure the state of the plant
2. Using the measured state, compute an open-loop optimal control law by optimising a cost function related to a predictive model.
3. Use control law as input for plant
4. Make new measurement and return to 2.

3.1.1 Mathematical formulation of MPC

Below, the open-loop optimal control problem will be formulated for the discrete case in accordance with Mayne et al. (2000). The dynamics of the plant that is to be control is determined by the difference equation:

\[ x(k+1) = f(x(k), u(k)), \quad (3.1) \]
\[ y(k) = h(x(k)), \quad (3.2) \]

with control, state and terminal constraints defined by

\[ u(k) \in U, \quad (3.3) \]
\[ x(k) \in X, \quad (3.4) \]
\[ x(k+T) \in X_f \quad (3.5) \]

where \( f(0,0) = 0; U \) is a convex, compact subset of \( \mathbb{R}^m \) such that \( 0 \in U; X \) is a convex, closed subset of \( \mathbb{R}^n \) such that \( 0 \in X; \) and the terminal time is defined by \( T = T_c = T_h. \)

A control sequence is denoted \( u(\cdot) \) or \( u \) and with time horizon \( T \) and terminal time \( k+T, \) it is defined by the sequence \( u = \{ u(k), u(k+1), \ldots, u(k+T-1) \}. \) Since the terminal time \( T \) is increasing with the current time \( k, \) the MPC is
The set of feasible control laws given the initial state $u$ sequence from the plant in eq. (3.1) when the control sequence $u$ is applied to the initial state $\hat{x}$ at time $k$ is denoted $x^u(\hat{x}, k)$ or, equally, $x^u(\hat{x}, k)$ and thus $x^u(k; \hat{x}, k) = \hat{x}$. To simplify notations, $x^u(i; (x, k))$ will usually be denoted $x(i)$. With this, the state trajectory can be written as $x^u(\hat{x}, k) = \{\hat{x}, x(k + 1), \ldots, x(k + T)\}$. An important set of control sequences is that containing those controls who ensures that the constraints in (3.3-3.5) are fulfilled. These are called the feasible controls, and they are defined below.

**Definition 1** (Feasible control). A control law $u = \{u(k), u(k + 1), \ldots, u(k + T - 1)\}$ for initial state $\hat{x}$ at time $k$ is called feasible (or admissible) if it satisfies the control, state and terminal constraints, i.e. if:

(i) $u(i) \in U$, $i = k, k + 1, \ldots, k + T - 1$

(ii) $x^u(\hat{x}, k) \in X$, $i = k, k + 1, \ldots, k + T - 1$

(iii) $x^u(k + T; (x, k)) \in X_f$.

The set of feasible control laws given the initial state $\hat{x}$ is denoted by $\mathcal{U}_T(\hat{x})$.

The cost function is given by a combination of a stage cost $\ell(x, u)$ and a terminal cost $F(x)$. Starting at state $\hat{x}$ at time $k$ and applying the control sequence $u$, the cost is given by:

$$V_T(x, k, u) = \sum_{i=k}^{k+T-1} \ell(x(i), u(i)) + F(x(k + T)), \quad (3.6)$$

where the index $T$ is added to $V_T(x, k, u)$ to clarify that the time horizon is $T$. To guarantee stability of the control law at a later stage, it is demanded that the stage cost satisfies $\ell(x, u) \geq c(\|x, u\|^2)$ and $\ell(0, 0) = 0$. Next, what is meant by an optimal control and a value function will be defined.

**Definition 2** (Optimal control and Value function). A feasible control law $u^0 = \{u^0(k), u^0(k + 1), \ldots, u^0(k + T - 1)\}$ for initial state $\hat{x}$ is optimal if $\forall u \in \mathcal{U}_T(\hat{x})$

$$V_T(\hat{x}, k, u) \leq V_T(\hat{x}, k, u^0).$$

The corresponding optimal cost $V_T(\hat{x}, k, u^0)$ is denoted $V_T(\hat{x}, k)$ and is called a value function.

The problem of finding the optimal control is denoted $\mathcal{P}_T(x, k)$ and using definition 2 it can be written as:

$$\mathcal{P}_T(\hat{x}, k) : V_T(\hat{x}, k) = \min_{u \in \mathcal{U}_T(\hat{x})} V_T(\hat{x}, k, u).$$

Only the first control $u^0(k; (x, k))$ from the optimal open-loop control law that solves $\mathcal{P}_T(x, k)$ will be applied to the plant. This will yield a new state of the plant $x^+$ and a new problem $\mathcal{P}_T(x^+, k + 1)$ that can be solved to generate yet another open-loop control law $u^0(x^+, k + 1)$ and corresponding first control $u^0(k + 1; (x^+, k + 1))$. A control sequence $\kappa^T(x, k)$ can then be defined implicitly by

$$\kappa^T(x, k) = u^0(k; (x, k)). \quad (3.7)$$
CHAPTER 3. THEORY

Observe that $\kappa_T(x,k)$ defines a closed-loop control law.

Since the stage cost $\ell(x,u)$, terminal cost $F(x)$ and $f(x,u)$ in eq. (3.1) all are time invariant, the value function $V^0_T(x,k) = V^0_T(x,0)$ and the problem $\mathcal{P}_T(x,k)$ becomes time invariant. The control law does thus only depend on the current state $x$ of the plant and not on the current time $k$. To simplify notations and to clarify discussions concerning general properties of MPC, the problem will hence be formulated without the index $k$ as:

$$\mathcal{P}_T(\hat{x}) : V_T(\hat{x}) = \min_{u \in U_T(\hat{x})} V_T(\hat{x},u),$$

with value function $V_T(\hat{x})$ and cost function $V_T(\hat{x},u)$ defined as before. The closed-loop control law can again be defined as $\kappa_T(\cdot) = u_0(0;x)$.

The ideal would be to find an explicit solution $\kappa_T(\cdot)$. For easy structures of the problem $\mathcal{P}_T(x)$, this can achieved with, for example, dynamic programming. However, in general it is impossible but as stated in Mayne et al. (2000), what differs MPC from dynamic programming is only the implementation, not the principle. In practise, what is done is thus to apply a feedback control $\kappa_T(x) = u_0(0;x)$ that is derived implicitly by solving the problem $\mathcal{P}_T(x)$ for the current $x$.

3.1.2 Stability of closed-loop control law

An important issue for the usability of the MPC is whether it can be guaranteed that the control law $\kappa_T(x)$ really steers the state to the origin, i.e., the stability of the solution. To prove that this is indeed the case, the value function $V_T$ will be employed as a Lyapunov function. The definition of a Lyapunov function and a theorem linking properties of this function with (asymptotical) stability will first be given.

Definition 3 (Lyapunov function). Let $X_0 \subset \mathbb{X}$ be a neighbourhood of zero. The function $V(x) : \mathbb{R}^n \rightarrow \mathbb{R}$ is a Lyapunov candidate function if:

(i) $V(0) = 0$
(ii) $V(x) > 0$, $\forall x \in X_0 \setminus 0$

i.e., if $V(x)$ is positive definite in a neighbourhood of zero.

Theorem 1 (Asymptotical stability). If $V(x) : \mathbb{R}^n \rightarrow \mathbb{R}$ is a Lyapunov candidate function, then $x = 0$ is asymptotically stable $\forall x \in \mathbb{X}$ if:

$$\Delta V(x) < 0, \forall x \in \mathbb{X} \setminus 0,$$

i.e., if $\Delta V(x)$ is negative definite for all $x \in \mathbb{X}$.

where $\Delta V(x(k)) = V(x(k)) - V(x(k-1))$.

There are some different ways to derive sufficient conditions for stability of MPC, and below the so called direct approach of the proof will be given. The proof becomes significantly easier if one has terminal equality constraints, i.e., if $F(x) \equiv 0$ and $X_f = \{0\}$. If this is not the case, a local controller $\kappa_f(x)$ is defined so that $X_f$ is invariant to $x(k+1) = f(x(k), \kappa_f(x(k)))$.

23
CHAPTER 3. THEORY

The notation $g(x, u)^*$ will be used to denote the change of value for a function $g(x)$ as the control $u$ is applied to the system, i.e., when the state moves from $x$ to $x^+ = f(x, u)$. Thus:

$$g(x, u)^* = g(x^+) - g(x).$$

Further, $X_k \subset \mathbb{X}$ is defined as the subset of $\mathbb{X}$ that is steerable through a feasible control sequence to the terminal set $X_f$ in at most $k$ time steps. Thus:

$$X_k = \{x \in \mathbb{X} \text{ s.t. } \exists u \in \mathcal{U}_k(x)\}$$

since the set $\mathcal{U}_k(x)$ is defined as those controls $u \in \mathcal{U}$ that steers the state $x$ to the terminal state $X_f$ within $k$ time steps. With these definitions, the sufficiency conditions for stability of MPC can be formulated below and the proof carried out (Mayne et al., 2000).

**Theorem 2** (Sufficient conditions for stability of MPC). The closed-loop problem defined by applying $\kappa_T(x, k)$ from eq. (3.7) to the plant in eq. (3.1)-(3.5) is stable if:

(i) $X_f \subset \mathbb{X}$, $X_f$ is closed and $\mathbf{0} \in X_f$.

(ii) $\kappa_f(x) \in \mathcal{U}$, $\forall x \in X_f$

(iii) $f(x, \kappa_f(x)) \in X_f$, $\forall x \in X_f$

(iv) $[F^* + \ell(x, \kappa_f(x))] \leq 0$, $x \in X_f$

Proof. Let $V_T(x)$ be the value function corresponding to $\mathcal{P}_T(x)$ as defined above where $x \in X_T$ for some time horizon $T$. The value function is given by applying the optimal control law $u^0(x) = \{u^0(0; x), u^0(1; x), \ldots, u^0(T - 1; x)\}$ to the problem and thereby reaching the optimal state trajectory given by $x^0(x) = \{x, x^0(1; x), \ldots, x^0(T; x)\}$.

Since $V_T(x^+) \leq V_T(x^+, u(x^+))$, any control law $u(x^+) \in \mathcal{U}(x^+)$ can be used to create an upper bound on $V_T(x^+)$ and thus an upper bound to $V_T(x^+) - V_T(x)$. Since $u^0(x) \in \mathcal{U}(x)$, the abbreviated control law $\{u^0(1, x), \ldots, u^0(T-1, x)\}$ would be feasible to the would remain inside the feasible region while steering the state $x^+$ to $x^0(T; x) \in X_f$ in $T - 1$ time steps. By condition (ii) and (iii), the local controller $\kappa_f(x^0(T, x))$ could be used to steer $x^0(T, x) \rightarrow f(x^0(T, x), \kappa_f(x^0(T, x))) \in X_f$ and thus the control sequence

$$u(x^+) = \{u^0(1; x), u^0(2; x), \ldots, u^0(T - 1; x), \kappa_f(x^0(T; x))\}$$

is feasible to the problem $\mathcal{P}_T(x^+)$. The corresponding state trajectory becomes

$$x(x^+) = \{x^+, x^0(2; x), \ldots, x^0(T; x), f(x^0(T; x), \kappa_f(x^0(T; x)))\}$$

and the cost becomes

$$V_T(x^+, u(x^+)) = V_T(x) - \ell(x, u^0(0; x)) - F(x(k + T))$$

$$+ \ell(x^0(T; x), \kappa_f(x^0(T; x))) + F(f(x^0(T; x), \kappa_f(x^0(T; x))))$$

$$= - \ell(x, u^0(0; x))$$

$$+ F^*(x^0(T; x), \kappa_f(x^0(T; x))) + \ell(x^0(T; x), \kappa_f(x^0(T; x)))$$

24
CHAPTER 3. THEORY

\[ a_l = (n_i, n_j) \]

Figure 3.2: A link \( a \) connecting the two nodes \( n_1 \) and \( n_2 \) and inflow \( u_l \); outflow \( v_l \); flow of pedestrians \( x_l \); and point queue \( \lambda_l \) for the link.

and by using condition (iv):

\[ V_T(x) - V_T(x^+, u(x^+)) \leq \ell(x, u^0(0; x)). \]

Since \( \ell(x, u^0(0; x)) \geq c((x, u)^2) \), asymptotic stability is proved. \( \square \)

An interesting special case is when the terminal constraint is defined by \( X_f = 0 \). If this is the case, the terminal cost is constant and can be put to zero, i.e. \( F(x(k + T)) = 0 \), and by definition \( \ell(0, 0) = 0 \). Condition (i) is then satisfied since \( \{0\} \in X \) and is closed. Since \( \kappa_f(0) = 0 \in U \), condition (ii) is also satisfied, and so is condition (iii) since \( f(0, 0) = 0 \in X_f \). Finally, condition (iv) is satisfied since both \( F(x) = 0 \) and \( \ell(x, \kappa_f(x)) = 0 \) for \( x \in X_f \). Thus, if the terminal constraint is \( X_f = 0 \), the problem is stable.

Observe that the assumption \( \ell(x, u^0(0; x)) \geq c((x, u)^2) \) could be relaxed, in accordance with theorem 1, and still guarantee stability.

3.2 Point queue model – the predictive model

In this section, the equations that determines the flow in a dynamic network flow model, namely the point queue (PQ) model, will be derived. The section will start by a discussion on the general setup and governing equations of a dynamic network flow model before arriving at the PQ-model. Following this, the model will be discretised in order to enable the formulation of a optimisation problem based on the model.

3.2.1 Continuous dynamic network flow models

The network will be defined by \( G(A, N) \) where \( N \) is the set of nodes and two nodes \( n, m \in N \) is connected by a directed link \( a = (n, m) \in A \). The number of pedestrians on link \( a \) at time \( t \) is modelled by the flow \( x_a(t) \). The number of pedestrians entering respectively leaving link \( a \) at time \( t \) is modelled by the inflow \( u_a(t) \) respectively outflow \( v_a(t) \). Flow conservation on a link thus gives:

\[ \dot{x}_a(t) = u_a(t) - v_a(t). \quad (3.9) \]

A single link and corresponding notations can be seen in figure 3.2

The flow entering a node must equal the flow leaving a node. The set of incoming links \( I_n \) and outgoing links \( O_n \) is defined as \( I_n = \{ a = (m, n) \in A \} \) and \( O_n = \{ a = (n, m) \in A \} \).
CHAPTER 3. THEORY

Figure 3.3: The sets $I_n$ and $O_n$ and corresponding inflow and outflow to the node $n$. The flow into the node must equal the flow out from the node.

and $O_n = \{(n, m) \in A\}$. The flow entering a node $n$ is the sum of the flow leaving links $a \in I_n$ and the flow leaving a node is the sum of the flow entering links $a \in O_n$. See figure 3.3 for a clarification. Pedestrians entering or leaving the network is modelled by source and sink nodes. The inflow to node $n$ from outside the network is given by the source variable $u_s^n(t)$ and the outflow from node $n$ is given by the sink variable $v_s^n(t)$. Thus flow conservation on a node for node $n$ gives:

$$\sum_{a \in I_n} v_a(t) - v_s^n(t) = \sum_{a \in O_n} u_a(t) - u_s^n(t) \quad (3.10)$$

A third equation governs how the flow propagates along the link, or the link dynamics. This is modelled by relating the outflow from a link to the inflow and number of pedestrians on the link. This relation can be expressed in a number of way. One alternative is to use the transit time $\tau_a(x_a(t))$, defined as the time it takes for a pedestrian entering link $a$ with flow $x_a(t)$ to cross the link:

$$\tau_a(t) = f(\Gamma_a, \Omega_a(t)) \quad (3.11)$$

and using the relation:

$$v_a(t) = u_a(t + \tau_a(x_a(t))). \quad (3.12)$$

Alternatively, $v_a(t)$ can be expressed by:

$$v_a(t) = g(\Gamma_a, \Omega_a(t)). \quad (3.13)$$

In both eq. (3.11) and eq. (3.13), $\Gamma_a$ is a vector containing information about parameters that affect the flow on link $a$; and $\Omega_a(t)$ is a vector containing information about current and previous values of inflow $u_a(t)$ and flow $x_a(t)$. How to find appropriate functions $f$ and $g$ is then the issue.

In Nie and Zhang (2005), three different approaches for modelling the link dynamics is given. The most basic one is called the M-N model after the creators Merchant and Nemhauser, and is based on eq. (3.13). In the M-N model, the outflow is only related to the number of pedestrians on the link, so:

$$v_a(t) = g(\Gamma_a, x_a(t)).$$
CHAPTER 3. THEORY

A problem with this model is that if flow enters an empty link, part of it will cross the link in zero time, something that is obviously unrealistic. However, if the conditions that are to be modelled do not change to fast with time, it is able to give realistic result. During an evacuation, the flow on a link is varying a lot with time and therefore this model does not suffice.

A second approach is given by the delay function (DF) link model, instead based on eq. (3.11) and eq. (3.12). The DF model overcomes the problem with the M-N model and seems promising in that the transit time $\tau_a(x_a(t))$ could be related to to the flow $x$ in accordance with fundamental diagrams. Unfortunately, it is hard to implement in practise. A first problem, that is more complex than one might think, is how to choose the function $f$. If this is not done with great care, the so called First-in-first-out property might not be satisfied. A second problem arises if the model is to be discretised. Since the delay time is dynamic and the time steps in a discretised model are discrete, a method for determining discrete delay times must be developed. This easily creates problems and it is especially not well suited for the formulation of an optimisation problem.

A third approach is given by combining a static delay $\tau_a(x_a(t)) = \alpha_a$ with the introduction of a point queue $\lambda_a(t)$ at the exit node. This approach, the PQ-model, will in the following be discussed in more detail.

3.2.2 Governing equations in PQ-model

An alternative way of modelling the link dynamics is given by the PQ-model (Li et al., 2000; Nie and Zhang, 2005). Instead of simply relating the outflow $v_a$ to the flow $x_a$ on a link, as in the M-N model, or by introducing a dynamic delay function, as in the DF model, the link dynamics is modelled by the combination of a static delay $\tau_a(x_a(t)) = \alpha_a$, related to the free flow travel time of the link, and a point queue $\lambda_a$, modelling the flow limitation during congestion. It is called a point queue since there is no spill back, and this queue gives the name of the model.

Congestion is modelled by introducing a maximum outflow from the link, given by the bottleneck capacity $M_a$. The idea is based on the observation that up to a certain threshold value of the traffic density, the traffic can move at free flow speed $v_{ff}$. When that threshold is reached, the speed decreases and the outflow cease to increase. It should be noted that a fixed upper bound $M_a$ – independent of the traffic density on the link – is unable to exactly model the congestion, since the flow decreases rather substantially at high densities, as can be observed in the fundamental diagrams in figure 2.1.

The delay $\tau_a(x_a(t))$ on link $a$ is modelled as a static parameter and defined by $\alpha_a$. If there is no queue on the link and the flow that reach the exit is less than $M_a$, then $u_a(t - \alpha_a) = v_a(t)$, just like in eq. (3.12).

The variable $\lambda_a(t)$ is defined as the flow “queueing” at link $a$ during time $t$. A queue will form if the more flow than $M_a$ attempts to exit per time unit. The flow that attempts to exit the link is the sum of the flow in the point queue, $\lambda_a(t)$, and the flow that entered the link $\alpha_a$ time units earlier, $u_a(t - \alpha_a)$. Thus the dynamics of the queue can be defined by

$$\frac{d\lambda_a(t)}{dt} = \begin{cases} 0 & \text{if } \lambda_a(t) = 0 \text{ and } u_a(t - \alpha_a) < M_a, \\ u_a(t - \alpha_a) - M_a & \text{otherwise} \end{cases} \quad (3.14)$$
CHAPTER 3. THEORY

where \( u_a(t - \alpha_p) - M_a \) is the difference between the inflow and outflow of the queue. The outflow \( v_a(t) \) from the link can be defined in a similar way:

\[
v_a(t) = \begin{cases} 
    u_a(t - \alpha_p) & \text{if } \lambda_a(t) = 0 \text{ and } u_a(t - \alpha_p) < M_a, \\
    M_a & \text{otherwise}.
\end{cases}
\] (3.15)

The dynamics of the PQ-model is given by eq. (3.14) and eq. (3.15) together with eq. (3.9) and eq. (3.10)

3.2.3 Discretising the PQ-model

To enable the formulation of an optimisation problem, or more exactly a linear programming (LP) problem, the PQ-model will next be discretised. The problem will be formulated as a finite (rolling) horizon optimisation problem with \( T = T_h = T/\delta_t \), where \( \delta_t \) is the time step size. Thus, \( T \) is the number of time steps from the present time to the time horizon. If the present time step is \( k \), the final time step becomes \( k + T \). However, the problem will be time invariant, so to simplify notations, \( k = 0 \) will be the current time. For link \( a \): the queue at the beginning of time period \( i \) is denoted \( \lambda_a(i) \) and the flow denoted \( x_a(i) \); the flow leaving respectively entering the link during time period \( i \) is denoted \( v_a(i) \) respectively \( u_a(i) \); the maximum outflow is \( M_a \); and the transit time in (even) time steps is \( \alpha_a \), so the transit time \( \tau_a = \alpha_a \delta_t \). The outflow from and inflow to node \( n \) during time period \( i \) is denoted \( v_s(i) \) respectively \( u_s(i) \).

Flow conservation on each link is changed from eq. (3.9) to:

\[
x_a(i + 1) - x_a(i) = u_a(i) - v_a(i).
\] (3.16)

Flow conservation for nodes is changed from eq. (3.10) to:

\[
\sum_{a \in I_n} v_a(i) - v_s_n(i) = \sum_{a \in O_n} u_a(i) - u_s_n(i).
\] (3.17)

The following equations can be derived from eq. (3.14) and eq. (3.15):

\[
\lambda_a(i + 1) = \lambda_a(i) + u_a(i - \alpha_a) - v_a(i)
\] (3.18)

and

\[
v_a(i) = \min(M_a, \lambda_a(t_{i-1}) + u_a(k - \alpha_a)).
\] (3.19)

Thus if \( \lambda_a(i - 1) + u_a(k - \alpha_a) < M_a \), then \( \lambda_a(i) = 0 \) as stated in eq. (3.14).

3.2.4 Formulation of a linear programming problem

Combining eq. (3.16-3.19), the dynamics of the network is defined and together they provide the constraints for an optimisation problem.

Since the question is how to choose the evacuation route for pedestrians, the control variable should relate to how pedestrians choose which link to enter after exiting the previous link. This amounts to choosing the inflow variable \( u_a \). If one also assumes that the outflow of pedestrians from a queue could be controlled, the outflow equation defined by eq. (3.19) can be relaxed to the two inequality constraints:

\[
v_a(i) \leq M_a
\] (3.20)

\[
v_a(i) \leq \lambda_a(i) + u_a(k - \alpha_a).
\] (3.21)
Observe that eq. (3.21) already is ensured by \( \lambda_a(k) \geq 0 \) together with eq. (3.18). It will be omitted in the later formulated optimisation problem.

Since no pedestrians will enter the system during an evacuation, the inflow \( u_n^a(i) = 0 \) except for when the system is initialised, something that will be discussed in detail in section 4.2.1. In that special case, \( u_n^a(0) \) will be given. If \( N_{sink} \) defines the set of sink nodes, then \( v_n^a(i) \geq 0 \) if \( n \in N_{sink} \) and \( v_n^a(i) = 0 \) otherwise.

Knowledge is further needed about the previous inflow to all links, since this will influence when flow reach the point queue. Thus \( u_i(i) \) is given for all \( i \in [-\alpha_a, -1] \). The initial flow and point queues at each link is further known, so \( x_a(0) \) and \( \lambda_a(0) \) are given \( \forall a \in L \).

How to define the cost function will be discussed in section 4.2.3. A general formulation includes a stage cost function \( \ell(x_a(i), \lambda_a(i), u_a(i), v_a(i), v_n^a(i)) \), and a terminal cost function \( F(x_a(T), \lambda_a(T)) \), giving:

\[
V_T(x, u) = \sum_{a \in L} \left( \sum_{i=0}^{T-1} \ell(x_a(i), u_a(i), v_a(i), v_n^a(i)) \right) + F(x_a(T), \lambda_a(T)).
\] (3.22)

By combining the constraints in eq. (3.16-3.18) and eq. (3.20) with the cost function from eq. (3.22) and the variable constraints discussed above, the optimisation problem can be formulated as the following linear programming problem:

\[
\begin{align*}
\text{minimise} & \quad \sum_{a \in L} \left( \sum_{i=0}^{T-1} \ell(x_a(i), u_a(i), v_a(i), v_n^a(i)) \right) + F(x_a(T), \lambda_a(T)) \\
\text{subject to:} & \quad x_a(i + 1) - x_a(i) = u_a(i) - v_a(i) \quad \forall a \in L \forall i \in [0, T - 1] \\
& \quad \sum_{a \in L} v_a(i) - v_n^a(i) = \sum_{a \in L} u_a(i) - u_n^a(i) \quad \forall n \in N \forall i \in [0, T - 1] \\
& \quad \lambda_a(i + 1) = \lambda_a(i) + u_a(i - \alpha_a) - v_a(i) \quad \forall a \in L \forall i \in [0, T - 1] \\
& \quad v_a(i) \leq M_a \quad \forall a \in L \forall i \in [0, T - 1] \\
& \quad u_a(i) \geq 0, \quad v_a(i) \geq 0 \quad \forall a \in L \forall i \in [0, T - 1] \\
& \quad x_a(i), \lambda_a(i) \geq 0 \quad \forall a \in L \forall i \in [1, T] \\
& \quad u_a(i) \text{ given} \quad \forall i \in [-\alpha_a, -1], \forall a \in L \\
& \quad x_a(0), \lambda_a(0) \text{ given} \quad \forall a \in L \\
& \quad v_n^a(i) \geq 0 \quad \forall n \in N_{sink} \forall i \in [0, T - 1] \\
& \quad v_n^a(i) = 0 \quad \forall n \notin N_{sink} \forall i \in [0, T - 1]
\end{align*}
\] (3.23)
Chapter 4

Constructing model

In the previous chapter, MPC and the PQ-model was described and an optimisation problem formulated. It was, however, not made clear how to construct a network that could model the pedestrian movement in a building complex, or how to choose the parameters in the PQ-model. Neither was it made clear how to apply a solution obtained from the PQ-model to the social force model simulation. In this chapter, these practical details will be made clear. Following this, it will be described how the optimal solution will be used in the simulation. But first, how to construct the social force model simulation will be described.

4.1 Modelling the social force model

This section describes the modelling of the social force model with anisotropy that will be used for the simulation. From eq. (2.3) the implementation can seem quite straightforward, but there were some technical issues that had to be solved. Since the implementation of the social force model will influence the result, an introduction to how the main problems were solved will be presented below.

When initialising the simulation, the number of pedestrian to place and where to place them must be specified. According to these specifications, the pedestrians are placed randomly with the constraint that $r_i - r_j > 1.2r$ for all pedestrians $i \neq j$. This constraint is needed to ensure that the initial force between pedestrians does not become too big.

4.1.1 Modelling the wall force

Remember the equation for the wall force on pedestrian $i$ from wall $W$:

$$f_{iW} = A_i e^{-\frac{r_i - d_{iW}}{pW}} n_{iW} + kg(r_i - d_{iW}) n_{iW} - kg(r_i - d_{iW})(v_i \cdot t_{iW}) t_{iW}. \quad (2.12)$$

Here, a wall is defined as the line segment connecting two points. If the corners of a wall is located at $p_1$ and $p_2$, it is defined as $W = \{p_1 + (p_2 - p_1)s \forall s \in [0, 1]\}$, as in figure 4.1. It should be noted that since only stepwise linear forms can be created with this definition, it is not well suited for the modelling of circular shapes. With this definition, the distance between the wall $W$, defined by the two points $p_1$ and $p_2$, and pedestrian $i$ located at point $r_i$ can in turn be defined...
as the minimum (euclidean) distance between \( r_i \) and the wall \( W \). To simplify calculations, a slight variation of this definition is used here. The distance \( d_{iW} \) is modelled as the shortest distance from pedestrian \( i \) onto the line of which wall \( W \) is only a segment. The coordinates of the orthogonal projection of pedestrian \( i \) on that line will be denoted \( r^W_i \). The orthogonal projection of a pedestrian onto the walls in a room is marked by crosses in figure 4.1. Observe that in figure 4.1, the orthogonal projection of the pedestrian onto the line intersecting both \( p_1 \) and \( p_2 \) does not lie on \( W_1 \). If this is the case, the wall force is set to zero. To model this, an indication function \( g_{iW}(r^W_i, W) \) is defined as:

\[
    g_{iW}(r^W_i, W) = \begin{cases} 
        1 & \text{if } r^W_i \in W, \\
        0 & \text{otherwise.}
    \end{cases} \tag{4.1}
\]

A door is defined as a section of a wall and can also be expressed as the line segment connected by two corner points \( c_1 \) and \( c_2 \), i.e., \( D = \{c_1 + (c_2 - c_1)s \mid s \in [0, 1]\} \). This section of the wall should not influence the pedestrians by any force, and if the orthogonal projection of a pedestrian onto a wall lies on the door on that wall, the wall force is zero. This is the case for the lower door in figure 4.1. In the same way as for walls, an indication function for a door \( D \subset W \) is defined as:

\[
    g_{iD}(r^W_i, D) = \begin{cases} 
        1 & \text{if } r^W_i \in D, \\
        0 & \text{otherwise.}
    \end{cases} \tag{4.2}
\]

A problem with this way of defining the door and wall is that it creates a discontinuity in the wall force close to the door. If a pedestrian during a time step move from a position very close to the door, where no force is present, to a position just next to the door, the new wall force can get very large. Especially since \( k \) and \( \kappa \) in eq. (2.12) are very large. To overcome this problem, if \( g_{iD} = 1 \), the two corner points of the door influence the pedestrian by a wall force in accordance with eq. (2.12) but with \( W \) exchanged to \( c_p \) for \( p = 1, 2 \) and with \( n_{iW}, d_{iW} \) and \( t_{iW} \) changed in appropriate ways.

### 4.1.2 Modelling the social force.

Modelling the social force is more straight forward than modelling the wall force. The main issue is to determine which pedestrians that should affect each other with forces. If \( P \) is used to notate the set of all pedestrians, then \( P_R(t) \subset P \) is the subset of the pedestrians that are located in room \( R \) at time \( t \). Further, \( P_D(t) \subset P \) is the subset of the pedestrians that have an orthogonal projection on door \( D \) at time \( t \). In this model, pedestrians are only influenced by other pedestrians that are either located in the same room, or, if the pedestrian has an orthogonal projection on a door \( D \), also have an orthogonal projection on door \( D \). The set of pedestrians influencing pedestrian \( p_i \) is denoted \( P_{p_i}(t) \) the above can be formulated as:

\[
    P_{p_i}(t) = \begin{cases} 
        P_D(t) \cup P_R(t) & \text{if } \exists D \text{ in } R \text{ s.t. } g_{iD}(r^W_i, D) = 1 \\
        P_R(t) & \text{else.}
    \end{cases} \tag{4.3}
\]
where $R$ is the room in which the pedestrian is currently located. Using this set, the total social force affecting pedestrian $i$ becomes:

$$f_{i,\text{social}}(t) = \sum_{p \in P_i(t)} f_{ij}(t)$$  (4.4)

with $f_{ij}(t)$ as in eq. (2.10).

### 4.1.3 Modelling the driving force

Except for how to determine the desired direction $e_0^i(t)$ in eq. (2.4), modelling the driving force is very straightforward. The desired direction is modelled by the introduction of a desired door $D_0^i$ and a desired point $p_0^i(t)$. The desired door is the door in the current room that the pedestrian is currently moving towards. How this door is determined is discussed in section 4.3. The line segment defining each door is discretised into ten points. Every time step, which of the ten points of the desired door $D_0^i$ that are closest to pedestrian $i$ is calculated, and this point becomes the new desired point $p_0^i(t)$.

### 4.1.4 Updating position

When updating the position of the pedestrians, the three forces described above are added together as in eq. 2.3) to get the acceleration $m_i \frac{dv_i}{dt}$. Using an euler-approximation, this can be written as:

$$F_{i,\text{tot}}(t) = m_i \frac{dv_i}{dt} \approx m_i \frac{v_i(t + \Delta t) - v_i(t)}{\Delta t},$$  (4.5)

where $F_{i,\text{tot}}$ is the sum of the forces affecting pedestrian $i$ at time $t$. Since the velocity at time $t$ is known to be $v_i(t)$, the velocity at time $t + \Delta t$ can be approximated by:

$$v_i(t + \Delta t) = v_i(t) + \frac{F_{i,\text{tot}}(t) \Delta t}{m_i}$$  (4.6)
Equally, the position of pedestrian $i$ is updated using the approximation:

$$ v_i(t) = \frac{dx_i}{dt} \approx \frac{x_i(t + \Delta t) - x_i(t)}{\Delta t} \quad (4.7) $$

that gives:

$$ x_i(t + \Delta t) = x_i(t) + v_i(t)\Delta t. \quad (4.8) $$

A problem with this way of updating the position arises when $\Delta t$ is not chosen sufficiently small. There is a risk that two pedestrians get very close to each other, leading to extremely high forces between them. There are two ways of handling this. Firstly, $\Delta t$ could be chosen very small, but this does not guarantee that no problem arise and will severely increase the computational demand. Secondly, $\Delta t$ could be decided adaptively. Here, $\Delta t$ is chosen according to:

$$ \Delta t = \min(t_0, \frac{c_t}{a_{max}}) \quad (4.9) $$

where $t_0$ is the standard update time; $a_{max} = \max_i \frac{F_{tot,i}(t)}{m_i}$ is the highest acceleration among all pedestrians; and $c_t$ is a constant determining the biggest allowed for change in velocity for any pedestrian in the simulation. The implications of this definition of $\Delta t$ on the velocity in eq. (4.6) is made clearer if the expression in eq. (4.9) is used, giving:

$$ v_i(t + \Delta t) - v_i(t) = \begin{cases} \frac{F_{tot,i}(t) t_0}{m_i} & \text{if } \frac{c_t}{a_{max}} \leq t_0 \\ c_t & \text{else} \end{cases} \quad (4.10) $$

By changing the values of these constants until acceptable behaviour is reached, $c_t = 0.5 \text{ m/s}$ and $t_0 = 0.1 \text{ s}$ is obtained.

When the position has been updated, it must be checked if the pedestrian is still in the same room, since this determines if the desired door should be updated, as well as which walls and pedestrians the pedestrian should be affected by. This is checked by calculating if the pedestrian has crossed the line defining its desired door.

### 4.2 Modelling the PQ-model

In this section, the modelling of the PQ-model is described. In section 3.2, the governing equations were derived. In this section, the characteristics of the network is adapted for pedestrian flow in a two dimensional environment. First, the construction of the network is described. Second, how to interpret and estimate the bottleneck capacity $M_a$ and transit time $\alpha_a$ is discussed. Third, a cost function is defined that minimises the remaining evacuation time and decreases the number of alternative optimal but impractical solutions, in a way that satisfies the stability conditions in theorem 2.

#### 4.2.1 Constructing network from layout

The nodes of the network can be divided into three different groups: source nodes, sink nodes and door nodes. The source nodes are introduced to model the initial distribution of the pedestrian. If $M_r$ pedestrians are initially located...
CHAPTER 4. CONSTRUCTING MODEL

in room \( r \), then the flow into the source node \( n^\text{source}_r \) at the first time step is \( M_r \). Mathematically this means that:

\[
u^*_n(i) = \begin{cases} M_r & \text{if } n^\text{source}_r \in N^\text{source} \text{ and } i = 0, \\ 0 & \text{otherwise} \end{cases}
\]

with \( u^*_n(i) \) as in eq. (3.17); with \( M \) as the number of pedestrians starting in the room; and with \( N^\text{source} \) as the set of source nodes. Every room where pedestrians are initially located has a source node and this source node is connected by a source link to the doors of that room. In figure 4.2, the links connecting \( n^\text{source} \) to \( n_1, n_2 \) and \( n_3 \) are examples of such links.

The door nodes represent a door in a specific room, so every door is represented by two nodes, again as in figure 4.2 where for example \( n_1 \) and \( n_4 \) are two door nodes representing the same door but in different rooms. The link connecting two such nodes are called a new room link; and two door nodes that represent doors located in the same room are connected by same room links. The links connecting node \( n_1, n_2 \) and \( n_3 \) are examples of same room links. A sink node is just a door node located on the outside of an exit door, thus connected by a new room link to the other door node of that door. If the door represented by \( n_1 \) is an exit, then \( n_4 \) would be a sink node. Mathematically, a sink node is characterised by the constraint:

\[
\begin{cases} v^*_n(i) \geq 0 & \forall i \text{ if } n \in N^\text{sink}, \\ v^*_n(i) = 0 & \text{otherwise} \end{cases}
\]

where \( v^*_n(k) \) is the outflow as defined in eq. (3.17) and \( N^\text{sink} \) is the set of source nodes.

The reason why two door nodes are used to model a single door is that pedestrians from many different directions are likely to leave the room using the same door. The door will limit their combined flow, and to model this there needs to be some kind of limit to the number of pedestrians leaving the door node. One could solve this problem by limiting the outflow \( \sum_{a \in O_n} u_a(i) + v^*_n(i) \) from that node, but this would also limit the possibility to redirect pedestrians to other doors within the same room. It is to overcome this problem that the new room link is created.

When entering a door, the pedestrian will enter a new room and be given a new desired door. It therefore make sense to make it impossible for pedestrians to stay at new room links, meaning that

\[
x_a(i) = 0 \quad \forall i \text{ if } a \in L^\text{new room link} \\
\alpha_a = 0 \quad \forall a \in L^\text{new room link}
\]

where \( L^\text{new room link} \) is the set of new room links.

4.2.2 Determining PQ-parameters

At the initial time it is hard to estimate where at a link pedestrians should be located. If a large number of pedestrians are located in the same room, it is likely that the doors of that room very soon will be used at the top of their capacity, namely the bottleneck capacity \( M_a \). The movement of pedestrians
Figure 4.2: Network construction from layout of a room with three doors. The node $n_{\text{source}}$ is an example of a source node and the links connecting $n_{\text{source}}$ to the door nodes $n_1$, $n_2$ and $n_3$ are called source links. The links connecting $n_1$, $n_2$ and $n_3$ to each other same room links and the link connecting $n_1$, $n_2$ and $n_3$ to $n_4$, $n_5$ and $n_6$ respectively are called new room links.

from their starting positions towards the doors in their first room will therefore only slightly influence the evacuation time. This means that setting the transit time $\alpha_a$ of the source links to zero will not create to big errors and will simplify the initial estimation of the states. It should be noted that this simplification badly captures the situation when a small group of people are located in a long corridor.

Since flow passing through either a source or a same room link eventually must pass through a new room link, and this link will have a smaller bottleneck capacity than either of the two other links, it will be the bottleneck capacity of the new room link that ultimately limits the flow of pedestrians on all links. Further, since there are no physical limitations to the number of pedestrians that can be redirected from one door to another door in the same room, the bottleneck capacity of source and same room links could be modelled as infinite.

There are many different parameters in the PQ-model that represent different physical quantities. What these represent and how they are estimated will be discussed below.

**Bottleneck capacity** $M_a$ The bottleneck capacity is the maximum amount of flow that can leave a link during a single time step. For a new room link it can be calculated directly from the physical property which it is supposed to model. With maximum specific flow $J_{\text{max}}$, width of door $w_a$ and time step size $\delta_t$, the bottleneck capacity is given by:

$$M_a = J_{\text{max}}w_a\delta_t. \quad (4.11)$$

Bottleneck capacity for different kinds of links is given in table 4.1.

**Transit time** $\alpha_a$ The transit time is the number of time steps it takes flow to reach the point queue after entering a link. With free flow speed $v_{ff}$ and with the length of the link $l_a$, the transit time $\alpha_a$ for a same room link is
CHAPTER 4. CONSTRUCTING MODEL

<table>
<thead>
<tr>
<th>Transit time $\alpha_a$</th>
<th>Bottleneck capacity $M_a$</th>
</tr>
</thead>
<tbody>
<tr>
<td>$L_{\text{same room}}$</td>
<td>$\text{nint} \frac{l_a}{v_{ffv} \delta t}$</td>
</tr>
<tr>
<td>$L_{\text{new room}}$</td>
<td>0</td>
</tr>
<tr>
<td>$L_{\text{source}}$</td>
<td>0</td>
</tr>
</tbody>
</table>

Table 4.1: Transit time and bottleneck capacity for the three kinds of links. Free flow speed for pedestrians is denoted by $v_{ffv}$; length of a link is $l_a$; maximum pedestrian flow is $J_{\text{max}}$; and width of a door is $w_a$ given by:

$$\alpha_a = \text{nint} \frac{l_a}{v_{ffv} \delta t} \quad (4.12)$$

where $\text{nint } x$ is the nearest integer to $x$. Transit time for different kinds of links is given table 4.1.

**Time step $\delta_t$** The size of the time step $\delta_t$ determines two properties of the control law. First, it determines the size of a time step in the PQ-model. Second, it determines how often a new control law is calculated. The size of $\delta_t$ in the PQ-model will thus not only determine how often the control can be updated, it will also determine how exact a real length $l_a$ can be represented by a transit time $\alpha_a$ since the transit time must be rounded as in eq. (4.12). A smaller $\delta_t$ will give a smaller error to this estimation of the transit time and is therefore necessary to obtain a network that correctly estimates the simulation. Here, $\delta_t = 2 \text{s}$ will be used.

**Number of time steps $T$** The number of time steps is chosen to guarantee that the flow in the PQ-model will reach zero before the last time step $T$. This is to guarantee that the stability conditions in theorem 2 are satisfied. In this report, $T = 50$ will be sufficient. With $\delta_t = 2 \text{s}$, $T = 50$ means that the time horizon becomes $T_h = 100 \text{s}$.

**Free flow speed $v_{ffv}$** The free flow speed is the speed of flow in the absence of congestion. The value of $v_{ffv}$ will influence the fundamental diagram of the PQ-model and can thus be used to calibrate the model against the social force model. This could be achieved either by comparing the flow through a single corridor; or by comparing the total estimated evacuation time in the PQ-model with the evacuation time in the simulation and change $v_{ffv}$ so that the two coincide. An alternative, less elaborate, method for choosing $v_{ffv}$ is to use the desired speed from the social force model $v^0$. This will be used here, so in accordance with section 2.2, $v_{ffv} = v^0 = 1.5 \text{ m/s}$.

**Maximum specific flow $J_{s, \text{max}}$** With the definition of $M_a$ as in eq. (4.11), the choice of $J_{s, \text{max}}$ will ultimately determine the congestion in the PQ-model. It could thus be calibrated together with $v_{ffv}$ to give realistic result. Here, $J_{s, \text{max}} = 1.8 / \text{m s}$ will be used.

To compare the PQ-model with the social force model, velocity $v^{PQ}$, specific flow $J_{s}^{PQ}$ and density $\rho^{PQ}$ must be defined from the link variables. To define
the velocity $v^{PQ}$, let $t_a^{PQ}(t)$ be the time spent on the link by flow entering at time step $t$. Then it makes sense to define the time $t_a^{PQ}$ by the combination of the transit time and queueing time:

$$\frac{t_a^{PQ}(t)}{\delta_t} = \alpha_a + \frac{\lambda_a(\alpha_a + t)}{M_a}.$$  

With this, the velocity on the link can be defined as:

$$v_a^{PQ}(t)\delta_t = \frac{M_a l_a}{\alpha_a M_a + \lambda_a(\alpha_a + t)}.$$  \hspace{1cm} (4.13)

What's left is now to relate the point queue $\lambda_a(\alpha_a + t)$ to the density $\rho_a^{PQ}$, and the specific flow will follow. To do this, first define the density by $\rho_a^{PQ}(t) = x_a(t) w_a l_a$ and observe that:

$$x_a(t) = \lambda_a(t) + \sum_{i=t-a+1}^t u_a(i).$$

Suppose that inflow $u_a(t)$ and flow $x_a(t)$ is constant. If $u_a < M_a$ then $\lambda_a = 0$ and $x_a = u_a \alpha_a < M_a \alpha_a$ and thus $\rho_a^{PQ} < \frac{M_a \alpha_a}{w_a l_a}$. If $u_a > M_a$ then $x_a$ will increase, so this case will not be considered. If $u_a = M_a$, then $x_a = \lambda_a + \alpha_a M_a$ and $\rho_a^{PQ} = \frac{\lambda_a + \alpha_a M_a}{w_a l_a}$. With $M_a$ as in eq. (4.11) and $\alpha_a$ as in eq. (4.12) where for simplicity $\text{nint}$ is overlooked, this gives:

$$\lambda_a = \begin{cases} 0 & \text{if } \rho_a^{PQ} < \frac{J_{s,max} \cdot v_{ffv}}{\rho_a^{PQ}} \\ \frac{M_a l_a}{\alpha_a M_a + \lambda_a(\alpha_a + t)} & \text{else} \end{cases}$$

and the velocity in eq. (4.13) becomes:

$$v_a^{PQ}(\rho_a^{PQ}) = \begin{cases} v_{ffv} & \text{if } \rho_a^{PQ} v_{ffv} < J_{s,max} \\ \frac{J_{s,max}}{\rho_a^{PQ}} & \text{else} \end{cases}.$$  \hspace{1cm} (4.14)

The specific flow is thus given by:

$$J_s^{PQ}(\rho_a^{PQ}) = \begin{cases} v_{ffv} \rho_a^{PQ} & \text{if } \rho_a^{PQ} v_{ffv} < J_{s,max} \\ J_{s,max} & \text{else} \end{cases}.$$  \hspace{1cm} (4.15)

The expressions obtained for $v^{PQ}$ and $J_s^{PQ}$ in eq. (4.14) and eq. (4.15) could have been obtained directly from the definition of $v_{ffv}$ and $J_{s,max}$. By definition, the speed of the flow is $v^{PQ} = v_{ffv}$ if there is no congestion, and there is congestion precisely when $\rho v_{ffv} \geq J_{s,max}$. If there is congestion, then $\rho v^{PQ} = J_{s,max}$, and from these simple observations, eq. (4.14) and eq. (4.15) follows.

### 4.2.3 Cost function

The cost function used in this report will minimise the remaining evacuation time without explicitly taking factors such as congestion or safety into consideration. The total remaining evacuation time is just the sum of $x_a(i) \cdot \delta_i$ for all time steps $i$ and all links $a$, where $\delta_i$ is the size of a time step. A problem with this cost function is that the number of alternative optimal solutions is
very large, with a lot of solutions corresponding to control strategies that are bad from a practical point of view. Consider once again the room in figure 4.2. If there are large point queues at the links ending at node \( n_1, n_2 \) and \( n_3 \) - large enough for the transit time \( \alpha_a \) to be smaller than the waiting time as a consequence of congestion - flow might be sent to link \( (n_1, n_2) \) at the same time as flow was sent back on \( (n_2, n_1) \) without effecting the total evacuation time. To avoid this problem, inflow to a link is coupled with a small cost \( c_u = 0.1 \). This gives the cost function:

\[
V_T(x,u) = \sum_{a \in L} \sum_{i=1}^{T} \delta_t x_a(i-1) + c_u u_a(i) \tag{4.16}
\]

where the stage cost thus is \( \ell(x_a,u_a) = x_a(i-1) + c_u u_a(i) \). Since, by choosing the time horizon \( T \) and time step size \( \delta_t \) appropriately, in practise, \( X_f = 0 \). The terminal cost can thus be defined by \( F(x,u) = 0 \). Since \( x_a \geq 0 \) and \( u_a \geq 0 \), so \( \ell(x_a,u_a) > 0 \) if \( x, u \neq 0 \), the stability constraints are fulfilled.

There might still be multiple solutions corresponding to the same basic control strategy. For example, if there is a long queue at a link further down the network, flow might be kept at a point queue even though \( \nu_{PQ}^a < M_a \), without prolong the total evacuation time.

4.3 Applying control

In this section, it will be explained how and what data is extracted from the simulation of the the social force model, how this data is used in the PQ-model and finally how the resulting open-loop optimal control law is applied on the simulation. The section is divided into three parts, discussing respectively how the initial movement of pedestrians is determined; how pedestrians choose which door to move to when entering a new room; and how pedestrians can be redirected within the same room. When the term flow is used, it is a sign that the PQ-model solution is discussed, and when the term pedestrian is used, it is a sign that the simulation of the social force model is discussed. The optimal flow will typically be non-integer values and are always rounded before applied as controls to the simulation.

**Initial position and source control.** As discussed previously in section 4.2.2, initially the only data that is assumed to be available is the number of pedestrians that are located in every room. The exact position of the pedestrians is not available and for simplicity the source links are therefore given \( \alpha_a = 0 \). If \( M \) pedestrians are initially located in room \( r \), the flow into the source node \( v_{PQ}^r \) of the room is \( M \) during time step \( k = 1 \). Thus \( v_{PQ}^{\text{source}}(1) = M \). Since no pedestrians enter the system at a later stage, this only happens during the first time step. Further, since this is the first time step, all flow in the network is initially zero, so \( x_a(1) = 0 \forall a \) and \( \lambda_a(1) = 0 \forall a \).

With this as initial conditions, the linear program formed by eq. (3.23) is solved. From the initial solution, two types of controls are extracted. The first control is called source control and determines the amount of pedestrians that should use respectively door in their starting room. The second control is called new room control and determines where the pedestrians should move when they
CHAPTER 4. CONSTRUCTING MODEL

enter a new room; this control will be discussed later. The source control is given by the inflow to the source links. Using figure 4.2 as an example, the source control is given by \( u(n_{\text{source}}, n_1)(1) \), \( u(n_{\text{source}}, n_2)(1) \) and \( u(n_{\text{source}}, n_3)(1) \). If \( M \) pedestrians are initially in the room, then the sum of these controls equals \( M \).

If \( u(n_{\text{source}}, n_1)(1) = M_{n_1} \), this means that \( M_{n_1} \) pedestrians should be directed towards \( n_1 \). This is achieved by setting the desired door, in the social force model, of the \( M_{n_1} \) pedestrians closest to the door represented by node \( n_1 \) to that door. If \( M_{n_2} \) pedestrians should be directed towards the door represented by \( n_2 \), then the \( M_{n_2} \) closest of the remaining pedestrians get a desired point located on that door.

**Entering a new room.** When a pedestrian enters a new door in the simulation, it must be assigned a new desired door. The control involved in this is called a new room control. The value of this control is given by the inflow to same room links. As an example, if the flow \( M \) passes through the new room link \( a_{\text{new room}} = (n_4, n_1) \) in figure 4.2 during time period \( t \) according to the solution of the linear programming problem, then \( u(n_4, n_1)(t) = M = v(n_4, n_1)(t) \) since \( a_{\text{new room}} = 0 \) and \( x_{\text{new room}} = 0 \). This flow can then either be directed to the link \((n_1, n_2)\) or the link \((n_1, n_3)\). A feasible solution would also be to send it back to the link \((n_4, n_1)\), but this will never be optimal since the cost function in eq. (4.16) includes a cost term for \( u \). The values of the inflow \( u(n_1, n_2)(t) \) and \( u(n_1, n_3)(t) \) indicates the amount of pedestrians that should be directed to respectively link. Since it is improbable that exactly \( M \) pedestrians enters the door in the simulation, the values of \( u(n_1, n_2)(t) \) and \( u(n_1, n_3)(t) \) cannot be used directly. Instead the expression

\[
u_{\text{prob}}^{(n_1, n_2)}(t) = \frac{u(n_1, n_2)(t)}{\sum_{n_k \in O_{n_1}} u(n_1, n_k)(t)} \tag{4.17}
\]

is used to determine the proportion of pedestrians that should be directed to respectively door. Here \( O_{n_1} \) denotes the set of outgoing links from node \( n_1 \). Every time a pedestrian enters a new room in the simulation, a uniformly distributed random variable \( X \sim U(0, 1) \) is used to determine its new desired door. This introduce a randomness to the simulation. The number of pedestrians that are directed to a new door in the simulation is saved and used as initial conditions for \( x, \lambda \) and previous values of \( u \) in the PQ-model.

Even if there is no flow into a room during the first time period of the optimal solution, it is possible that pedestrians in the simulation will enter that room. If so, the expression in eq. (4.17) does not make sense. To handle this discrepancy between the simulation and the simplifying PQ-model, it is for all door nodes checked if \( \sum_{n_k \in O_{n_1}} u(n_1, n_k)(t) > 0 \). If so, eq. (4.17) is used directly. Otherwise, the sum is checked for a specified number of the following time steps, to see if any of them is greater than zero. If so, the first of these is used as control. Otherwise, no control is used. The number of time steps to check is chosen to four and this is high enough to not cause any problems.

**Changing desired door within the same room.** The third possible control is called same room control and gives the opportunity to redirect pedestrians to other doors within the same room. For example, it is possible that many more pedestrians than predicted by the PQ-model was able to leave the door at node
Therefore pedestrians waiting at the door at node \( n_3 \) should be redirected towards the door at node \( n_1 \). It is not possible to derive from which link pedestrians should be redirected using the PQ-model, only the number of pedestrians to redirect. The number of pedestrians to redirect within the room is given by the total flow entering the new link minus an estimation of the flow entering the link from outside the room based on eq. 4.17. The estimation of the flow \( u_{n(n_i, n_j)}^{\text{same room}}(t) \) that is redirected to a same room link \((n_i, n_j)\) is given by:

\[
u_{n(n_i, n_j)}^{\text{same room}}(t) = u_{(n_i, n_j)}(t) - u_{(n_i, n_j)}^{\text{prob}}(t) v_{(n_k, n_i)}(t) \] (4.18)

where \( n_k \) is connected to \( n_i \) by a new room link and the flow entering \( n_i \) from this link is \( v_{(n_k, n_i)}(t) \).

Pedestrians are redirected to the link \((n_i, n_j)\) from all same room links that ends in \( n_i \). If \( u_{n(n_i, n_j)}^{\text{same room}}(t) = M \), then the \( M \) pedestrians that are closest to the door at node \( n_j \) but are currently moving towards the door at node \( n_i \) are redirected towards node \( n_j \). For the initial conditions of the linear programming problem, flow is modelled to have entered the link \((n_i, n_k)\) during time step \( t \), independently of where in the room the pedestrians were positioned when redirected. This means that the time it will take for redirected pedestrians to reach their new desired door typically will be overestimated in the PQ-model.
Chapter 5

Calibration and simulation layout

This chapter will start by describing the scenario used for calibrating the social force model. This will be combined with a discussion of how to compare the social force model with the PQ-model and to theoretical fundamental diagrams. Following this, the scenarios used for testing the evacuation strategy is described and parameters used in the simulation experiments is given.

5.1 Calibration of social force and PQ-model

The social force model will be calibrated to fit with the fundamental diagram as presented in figure 2.1, although with slightly different values of the constants. Calibrate here means that the constants $A$ and $B$ in the social force described by eq. (2.10) will be adapted so that the flow will increase and decrease as expected. Constants governing physical forces are used directly from literature, as described in section 2.2. The constant $A$ and $B$ will be calibrated to give a maximum flow $J_{s,max} = 2 / \text{m s}$ and zero flow for $\rho_0 = 6 / \text{m}^2$. As discussed in section 2.1, empirical values of these parameters vary immensely. The values chosen for calibration here is vaguely based on the fundamental diagram presented in Yu and Johansson (2007). There are infinitely many $A$ and $B$ combinations that give the same fundamental diagram. Therefore one of the constant could be chosen on some other ground. Here, $B = 1 \text{ m}$ is used for simplicity.

A problem with calibrating the flow as a function of pedestrian density is how one should measure the density. To simplify this, a straight corridor with periodic boundary conditions is used, so the number of pedestrians is constant and approximately uniformly distributed in the corridor. Here, periodic boundary conditions means that the two ends of the corridors are connected to each other. With $N_{\text{ped}}$ as the number of pedestrians in the corridor, $l_c$ as the length of the and $w_c$ as the width, the medium density $\tilde{\rho}$ is defined as:

$$\tilde{\rho} = \frac{N_{\text{ped}}}{l_c w_c} \quad (5.1)$$

Since the width of the corridor is constant, the medium specific flow $\bar{J}_s$ in the
The corridor used in calibrating the flow can be seen in figure 5.1. The width of the corridor is \( w_c = 3 \text{ m} \) and the length \( l_c = 20 \text{ m} \).

To compare the social force model with the PQ-model in terms of flow, the velocity \( v_{PQ} \) and specific flow \( J_{PQ} \) from eq. (4.14) respectively eq. (4.15) will be used.

### 5.2 Verification of control law

In order to test the control law, a couple of simple scenarios are created. In the first scenario, only one exit and two possible routes leading to the exit is created. Different control strategies is thus equal to directing different amount of pedestrians to respectively route. In the second scenario, an extra exit is created, giving more flexibility to the evacuation.

#### 5.2.1 Scenario 1

The layout used when testing the model is given in figure 5.2. Initially, 400 pedestrians are placed in room \( R_1 \). There is only one exit from the area, namely door \( D_6 \). While in \( R_1 \), the pedestrians can choose to use either \( D_1 \) or \( D_2 \). The control variable in this scenario is thus which door to use in \( R_1 \).

To evaluate how well the PQ-model combined with MPC is able to provide effective route guiding and handle deviances, three different evacuation strategies are tested.

**Closest door** In the closest door strategy, pedestrians always chose the door closest to their current position that leads towards the exit. This strategy is used for reference.

**PQ-strategy** This is the strategy that are evaluated and compared to the closest door strategy. Pedestrians choose their route according to the PQ-model and MPC combination.

**PQ-strategy with error** The third strategy is called PQ-strategy with error. The routing is calculated in the same way as in the PQ-strategy, except that some errors in the network approximations leads to non-optimal solutions. In this example, \( D_1 \) is set to \( w_{error}^{D_1} = 1 \text{ m} \) (marked with blue
5.2 Scenario 2

In scenario 2, the size of door $D_6$ is changed to $4 \text{ m}$. Further, $D_7$ is added to room $R_5$. The width of this door is only $w_{D_7} = 1 \text{ m}$ and therefore all pedestrians leaving $R_1$ through $D_1$ will not be able to use $D_7$. Pedestrians entering $R_5$ through $D_6$ should therefore be divided between $D_6$ and $D_7$. How the pedestrians are divided could be expected to be dependent on the time and the number
of pedestrians still in the system. One could for example expect that the last pedestrians entering $R_5$ through $D_5$ should all use $D_7$.

Only the PQ-strategy will be used for scenario 2.
Chapter 6

Result

6.1 Calibration result

Figure 6.1 shows the medium velocity $\bar{v}(\rho)$ of pedestrians from a simulation of the corridor in figure 5.1. Corresponding fundamental diagram is given in figure 6.2. In the figure, the medium of the result from ten different simulations are presented. Theoretical values for the PQ-model in accordance with eq. (4.14) and eq. (4.15) are included as well as Weidmann’s and Greenfield’s models from eq. (2.2) respectively eq. (2.1).

The social force model has been calibrated in order to give a maximum flow $J_{s,max} = 2/m \cdot s$ and zero flow for $\rho_0 = 6/m^2$. The data in figure 6.1 and 6.2 shows the result for $A = 29 N$ and $B = 1 m$. In figure 6.1, it is clear that the velocity decreases almost linearly with density. It is therefore in very good agreement with Greenfield’s model that are based on exactly this assumption. There is thus a lack in agreement with Weidmann’s model as can be seen in figure 6.2, where the social force model can be observed to overestimate the flow in comparison to Weidmann’s model. The maximum flow also becomes $J_{s,max} = 2.1/m \cdot s$ but no combination of $A$ and $B$ was found that could give both $\rho_0 = 6/m^2$ and $J_{s,max} = 2.1/m \cdot s$ at the same time. Since no thorough search for such combinations were done, it cannot be concluded that no such combinations exists. It might further be possible to get better result if by considering different values of the anisotropy factor $\lambda$ in eq. (2.6) and the physical force constants $k$ and $\kappa$ in eq. (2.10).

The constant $v_{ffv}$ and $J_{s,max}$ in the PQ-model is the same as those used to calibrate the social force model. This means that $J_{s,max} = 2/m \cdot s$ and $v_{ffv} = 1.5 m/s$. The velocity does not decrease initially in the PQ-model and it never reach zero. It can therefore be observed in figure 6.2 to give a higher flow than the social force model for small $\rho$ close to $\rho_0$. Overall, the fundamental diagram for the PQ-model resembles that from the simulation and therefore no further calibration of the PQ-model is considered necessary at this stage.

6.2 Simulation result

Below, simulation result for scenario 1 and scenario 2 will be given in the form of momentary figures from the simulation.
CHAPTER 6. RESULT

Figure 6.1: Dependence of velocity on density from simulations of the social force model in the corridor from figure 5.1. For every $\rho$, ten simulations were done and the medium is given here. Analytical expression is included for Weidmann’s model as in eq. (2.2), Greenfield’s model as in eq. (2.1) and the PQ-model as in eq. (4.14). The constants $A = 29$ N and $B = 1$ m is obtained after calibration to give a maximum flow $J_{s,\text{max}} = 2$ m/s and zero flow for $\rho_0 = 6$ m$^2$. In Wiedmann’s, Greenfield’s and the PQ-model, $v_{ff} = 1.5$ m/s, $J_{s,\text{max}} = 2$ m/s, $\rho_0 = 6$ m$^2$ and $c_w = 1.913$ m$^2$. 

\[ \rho \left[ \frac{1}{m^2} \right] \]

\[ \bar{v} \left[ m/s \right] \]
CHAPTER 6. RESULT

Figure 6.2: Fundamental diagram from a simulation of the social force model in the corridor from figure 5.1. For every $\rho$, ten simulations were done and the medium is given here. Analytical expression is included for Weidmann’s model as in eq. (2.2), Greenfield’s model as in eq. (2.1) and the PQ-model as in eq. (4.15). The constants $A = 29 \text{ N}$ and $B = 1 \text{ m}$ is obtained after calibration to give a maximum flow $J_{s,max} = 2/\text{m s}$ and zero flow for $\rho_0 = 6/\text{m}^2$. In Wiedmann’s, Greenfield’s and the PQ-model, $v_{ffv} = 1.5 \text{ m/s}$, $J_{s,max} = 2/\text{m s}$, $\rho_0 = 6/\text{m}^2$ and $c_w = 1.913 \text{ m}^2$. 
CHAPTER 6. RESULT

6.2.1 Scenario 1

Figure 6.3 - 6.5 shows the evacuation process for scenario 1 and the three strategies presented in section 5.2.1. Blue pedestrians are initially moving towards $D_1$ and red towards $D_2$. From figure 6.3 it is clear that to many pedestrians use $D_1$ in $R_1$ in the closest door strategy, since exactly half of the pedestrians initially placed in $R_1$ use respectively door. In figure 6.4, when the PQ-strategy is used, less pedestrians initially choose $D_1$ and the last blue pedestrians leave the room almost at the same time as the last red pedestrians. The possibility to redirect pedestrians in $R_1$ is not used.

In figure 6.5, the PQ-model underestimates the possible outflow from $D_1$, and therefore the blue group is considerably smaller here than in figure 6.4. However, as people leave $D_1$, the optimal solution changes and pedestrians are redirected from the red group to the blue group. Pedestrians that change group are green in the figures for scenario 1. Finally, the pedestrians that used $D_1$ and the ones that used $D_2$ leaves the exit at approximately the same time.

In figure 6.6, the difference between the three strategies becomes clear. The closest door strategy takes approximately 10 s longer than the PQ-strategies. In the PQ-strategy with error, the combination of the green and blue group is very close to the blue group for the PQ-strategy without error.

6.2.2 Scenario 2

Figure 6.7 shows the evacuation process for scenario 2 when the PQ-model is used to determine the route choice. When the new door is added, pedestrians are redirected without seemingly purpose between the two doors in $R_1$. Approximately 5-10 pedestrians are redirected between these two doors during the evacuation.

The first pedestrians that enters $R_5$ through $D_5$ are all directed towards $D_7$. When the flow through $D_5$ increases, approximately half of the pedestrians are directed towards $D_6$. In that way, no clogging occur at neither $D_7$ nor $D_6$.

When the last pedestrians enters $R_5$, all are routed towards $D_7$. This creates a small queue at $D_7$. 

CHAPTER 6. RESULT

Figure 6.3: Closest door strategy. The evolution of the evacuation when pedestrians choose the door closest to their position.
Figure 6.4: Simulation result for PQ-strategy (see section 5.2.1). Pedestrians that has been redirected would have been green, but this is never needed.
Figure 6.5: Simulation result for PQ-strategy with error on scenario 1 (see section 5.2.1). Green pedestrians have been redirected from $D_2$ to $D_1$ since the flow through $D_1$ is higher than predicted.
CHAPTER 6. RESULT

Figure 6.6: Number of red, blue and green pedestrians as a function of time for the three different strategies in scenario 1.
Figure 6.7: Scenario 2. An extra exit is added. Initially, all pedestrians that enters $R_5$ through $D_3$ use $D_6$, and when the flow increases, more are directed towards $D_6$. When the last pedestrians use $D_5$, all are directed towards $D_7$. Pedestrians that enters the $R_5$ through $D_3$ always use $D_6$. 
Chapter 7

Discussion

The two main issues that will be discussed here are whether the control law is able to provide efficient solutions and if it is able to handle errors in the data.

From the comparison of the three strategies for scenario 1, in figure 6.6, it is clear that the PQ-model manages to provide a better solution than the closest door strategy. It is thus clear that it would be better to use the PQ-model than to simply guide pedestrians to the closest door during an evacuation.

In figure 6.4, it can be observed that the last pedestrians entering $R_5$ from respectively door will reach $D_6$ at approximately the same time. This is what one could expect from an optimal solution since this indicates that the flow through $D_6$ is maximised. For this basic scenario, pedestrians usually do not change desired door in $R_1$, something that further indicates that the PQ-model is able to predict the outflow with some precision. Otherwise, pedestrians would have been moved between the two groups as in the PQ-model with error strategy.

In figure 6.5, pedestrians are redirected between the two doors in $R_1$. This is because the strategy used (PQ-model with error) underestimates the possible flow through $D_1$. When the data is updated, less pedestrians that expected and thus optimal are queueing at $D_1$, even with the low expected flow. Consequently, pedestrians in $R_1$ are rerouted from $D_2$ to $D_1$. The final evacuation time for this strategy is almost identical to the PQ-strategy without error, as is the total number of pedestrians that use $D_1$ respectively $D_2$. This indicates that the model is able to handle some errors in the predicted flow. To big conclusion should not be drawn from this, and more research is needed to determine how robust the model really is. Further, the ideal would be if the values used for determining $M_a$ in the PQ-model was updated according to data from the simulation. If so, all pedestrians could have been redirected as soon as it was realised that the maximum flow through $D_1$ was underestimated.

One problems with this way of testing the robustness is that the flow is underestimated for a door in the initial room. If wrong knowledge would have been given about $D_5$, the PQ-model might not have handled it equally well.

From the result for scenario 2, presented in figure 6.7, the model seems to be able to handle slightly more difficult scenarios as well, and it also shows why a statically defined control law is bad. However, in this scenario, pedestrians are redirected between the two doors in $R_1$. Since the redirection is in both ways, this indicates that it might not be due to an error in the PQ-model, since such an error should be almost constant throughout the simulation and would thus only
have rerouted pedestrians in one direction. The rerouting could therefore be to actual fluctuations in the flow in the simulation. The movement of pedestrians is to some extent chaotic and the flow through a door is not constant and for this more complicated scenario, the flow might fluctuate more than in scenario 1.

Some aspects of the evacuation process in scenario 2 are especially worth mentioning. First, when the first pedestrians reach $R_5$ through $D_5$, all of them move towards $D_7$. This is because the flow is initially so small that no queue occur at $D_7$ even if all pedestrians are routed there. Second, when the flow through $D_5$ has increased beyond the capacity of $D_7$, some pedestrians are instead routed towards $D_6$. Since $D_6$ is wide enough for all pedestrians from $D_5$ and $D_3$ together, no queue will occur there. Third, when only a few pedestrians remains, it takes shorter to have them queueing at door $D_7$ than to route them towards $D_6$. At $t = 50$ s, in figure 6.7, no more pedestrians are routed from $D_5$ to $D_6$, and a small queue is created at $D_7$. The most efficient solution, if there is a lot of pedestrians left in the system, becomes the one where $D_7$ is used to exactly its maximum capacity by pedestrians from $D_5$, and the rest use $D_6$. Observe that if all pedestrians were to choose the route with shortest evacuation time for them, a queue would occur at $D_7$ that took as long time to pass as the distance between $D_7$ and $D_6$ takes to walk.

These aspects highlights the pros of a dynamic escape route, dependent on data regarding the pedestrian distribution. When using the PQ-model strategy on scenario 1, a static evacuation plan based on the initial solution would have been sufficient. However, in scenario 2 the route choice is dependent on the number of pedestrians still left in the system, and a static escape route would not have been able to use $D_7$ in an optimal way.

### 7.1 Future work

A main issue is how well the control law is able to handle uncertainties, and even though the result in scenario 1 indicates that it is able to handle them to some degree, more tests are needed. In these simulations, all pedestrians follow the route assigned to them and the data used in the PQ-model corresponds exactly to that in the simulation. In reality, it is unlikely that all pedestrians would follow the assigned routes and it would not be possible to know towards which door every pedestrian was moving. How such errors would influence the control has not been tested.

A possible way to handle this would be to reformulate the linear programming problem in eq. (3.23) as a stochastic programming problem. The network flow characteristics of this linear programming problem makes it possible to solve it very efficiently, and it might therefore be possible to add stochastics without making it to computationally demanding.

In the cost function in eq. (4.16), only the total evacuation time is minimised. As discussed before, a better cost function should take more factors into account, such as the safety situation on different routes. A simple way of doing this would be to add a factor in front of $x_a$, dependent on the current safety of link $a$. A high cost factor would prevent pedestrians from using that link. The evacuation strategy presented in this thesis could thus easily be combined with a model describing, for example, the evolution of a fire.
CHAPTER 7. DISCUSSION

The closest door strategy is not a very realistic alternative strategy, so a comparison between this and the PQ-strategy is not a good way to estimate the time saved by using the PQ-strategy. A more realistic alternative could be attained in a couple of ways. A shortest-path strategy could easily be calculated from the network constructed for the PQ-model. It would also be interesting to compare the result with some kind of individual shortest time strategy, where every pedestrian was modelled as a time minimiser.


BIBLIOGRAPHY


