Traffic Flow on Escalators and Moving Walkways:
Quantifying and Modeling Pedestrian Behavior in a Continuously Moving System

Peter D. Kauffmann

Thesis submitted to the faculty of the
Virginia Polytechnic Institute and State University
in partial fulfillment of the requirements for the degree of

Master of Science
In
Civil Engineering

Dr. Shinya Kikuchi
Dr. Antoine Hobeika
Dr. Ralph Buehler

February 4, 2011
Falls Church, Virginia

Keywords: escalator, moving walkway, cellular automata, pedestrian, capacity

Traffic Flow on Escalators and Moving Walkways: Quantifying and Modeling Pedestrian Behavior in a Continuously Moving System

By
Peter D. Kauffmann

ABSTRACT

Because of perceived deficiencies in the state of the practice of designing escalators and moving walkways, a microsimulation-based model of pedestrian behavior in these moving belt systems was created. In addition to implementing walking and stair climbing capabilities from existing pedestrian flow literature, the model utilized following behavior and lane change decision logic taken form studies performed in the field of automotive traffic flow theory. By combining research from these two normally independent fields with moving belt operational characteristics, a solid framework for the simulation was created.

The model was then validated by comparing its operation to real world behaviors and performance metrics found in the literature in order to verify that the simulation matched the choices made by actual pedestrians. Once this crucial function had been completed, the model could finally be used in its original purpose of determining the capacity of a belt under region-specific input parameters. This paper also discusses other applications for which the model is suitable, including performing sensitivity analysis of both existing and proposed belt systems, analyzing the impacts of operational rule sets on the performance of escalators and moving walkways, and analyzing the effect of queue growth on the storage area needed for pedestrians in an ambulatory facility. Through the use of this model and the logic contained within it, engineers and planners will be able to gain a more accurate understanding of pedestrian flow on moving belts. The result of this increased understanding will be more effective and more efficient transportation systems.

Acknowledgements

I would like to first thank my advisor, Dr. Shinya Kikuchi, for his continued support and encouragement throughout my graduate career. His courses have opened my eyes to the underlying issues that face the field of transportation engineering in this country and around the world, and outside of class he has provided me with the opportunity to travel abroad and gain an international perspective on engineering issues. Without his advice on this project, I would likely never have found such a unique and engaging thesis topic, and his comments have helped it become a much stronger document.

I would also like to single out the members of my thesis committee for their invaluable support in this process. It wasn't always easy to schedule meetings of the three committee members – especially considering they work out of three separate Virginia Tech campuses – but because of their flexibility, we made it work. I thank Dr. Antoine Hobeika for advising me on the "how" of the model, ensuring that its operation was based in sound logic, and I thank Dr. Ralph Buehler for making sure that I remain mindful of the "why" of the project, thereby keeping the purpose and goal of the model in the forefront of my thoughts.

Also at Virginia Tech, I would like to thank the faculty of the Transportation and Infrastructure Systems group. I would like to especially call out the professors who assigned significant term projects. Although they seemed overwhelming at the time, I now understand that including large projects in the TISE coursework helps to prepare students for larger research undertakings, so thank you Dr. Montasir Abbas and Dr. Hesham Rakha, as well as Dr. Hobeika and Dr. Kikuchi.

My continued gratitude goes out to the Via Department of Civil and Environmental Engineering staff. Lindy Cranwell, Merry-Gayle Moeller, Val Dymond, Kara Lattimer, Donna Sanzenbach, Jeny Beausoliel, and the rest of the gang – you all were always there for me when I needed something, and for that, I will always be indebted to you. A big thanks goes out to former staff members Shelia Collins and Vickie Mouras as well.

Furthermore, I would like to thank Dr. Randy Dymond and Prof. Jeff Connor for giving me a way to support myself financially during the time I should have been working on my thesis. Teaching Measurements was a lot of fun, and I learned a lot about what it is like to be on the other side of the assignment dropbox. Your continued advice and support really aided me in my development as a student and as a teacher. Additional thanks goes out to Scott Miller and Kyle White, the two best student teaching partners I could have asked for.

Acknowledgement is also owed to my roommates Greg Swieter, Brad Shapiro, and Jeremy Henry, as well as my friends Kevin Finelli, Mallory Brangan, and Elaine Huffman for motivating me to go traveling while working on my thesis. Traveling through big airports and using public transit systems played a large role in forming the approach I took to writing this document.

Finally, I'd like to thank my parents for never once asking impatient questions about when I was going to finish up my graduate study and get a real job, and I'd like to thank my girlfriend for moving to France for a year so I finally had an excuse to sit down and get to work on this project.

You all are the best.

Table of Contents

Table of Figures

Chapter 1 – Introduction to Moving Belts

In the early days of urban transportation, achieving mobility was not difficult. Travelers could, given enough time, walk to whatever destination they needed to reach. If necessary, a horse or carriage might be used. However, as human settlements have become more and more dispersed, it has become necessary to develop new technologies to speed up travel within urban areas. In contemporary society, cities require tremendous transportation networks to handle the needs of their citizens. However, even with this reliance on mechanized modes like automobiles, buses, and trains, the need to accommodate pedestrians remains paramount.

In many public facilities like transit centers, airports, and even shopping centers, conglomeration of modes have occurred in order to gain the efficiencies of scale. Because of this, these structures have become so large that their designers cannot reasonably expect the average user to travel across the facility by means of walking alone, at least not in a timely manner. To this end, engineers and inventors have spent the last century developing innovations that can speed up pedestrians and facilitate movement through urban interface areas.

1.1 Prevalence of Escalators and Moving Walkways

Today, escalators and moving walkways have become an integral part of the urban aesthetic. Travelers have come to expect the presence of these enabling devices in most if not all significant public facilities. Escalators have become commonplace wherever elevation change is present, both to allow pedestrians to traverse a longer distance than normal even when carrying luggage and also to keep the flow of travelers at a stable and high rate even in the presence of vertical obstacles. Similarly, moving walkways have seen widespread implementation in dispersed facilities like airports, which are spread out by necessity, or adjacent transit stations, which may be connected to facilitate transfers between lines.

Both devices provide several key benefits. First, they serve to reduce overall travel time across a facility. At the same time, moving belt systems also increase both the horizontal and vertical distances that pedestrians are able to traverse by reducing the level of physical effort that must be expended. Finally, and perhaps most importantly in the context of a high-traffic area like a transit station, the presence of an escalator or moving walkway improves the overall flow through an otherwise constricted area, thereby reducing the area required in the facility to accommodate a given number of people. By conveying all passengers along at a constant rate, the belt also creates some minimum speed at which all riders must travel. The conveyance of the belt both serves to increase throughput while at the same time decreases the speed differential through the constriction, therefore reducing conflicts.

Despite the benefits that escalators and moving walkways provide, their numbers are limited by virtue of the limited number of public facilities where their usage is required. While it may be true that in high capacity, low-rise facilities there are few alternatives to escalators for movement, in the vast majority of hospitals, office buildings, and apartment buildings this purpose is fulfilled by the elevator. To this end, while the 30,000 escalators in operation in the United States (Slaughter, 2004) may seem like a large quantity, this number pales in comparison to the 700,000 elevators in operation (Hession). However, while there may have been over twenty times as many elevators, they only carried slightly more passengers. In the US and Canada, there are 325 million elevator passengers per day compared to 245 million escalators passengers per day (Hession). Therefore, it can be seen that despite their relatively limited number, escalators and moving walks on average carry a much greater load per unit.

1.2 History and Development of Moving Belt Systems

The development of moving belt technology was spread out over several decades, and indeed still continues to this day. The first patent for such technology on record was filed in August of 1859 by Nathan Ames for an invention he called "revolving stairs." This device, while crude and immensely dangerous by modern standards, was the first to include a progression of horizontal step-like surfaces – called "cleats" – for standing (Ames, 1859). Earlier models had instead consisted of a simple belt with regularly spaced wooden slats to provide traction (Lampugnani, Hartwig, Simmen, & Imorde, 1994).

However, the first modern version of the escalator as it is known today was premiered by the Otis Elevator Corporation at the Paris Exposition in 1900. Practically all contemporary escalators follow the model laid out by Otis. Safety features that were introduced in this model include side balustrades for support. However, the most noteworthy innovation of the Otis escalator was the inclusion of a number of flat steps at the entry and exit points to facilitate loading and unloading (O'Neill, 1974). By adding in space for two – and later three – consecutive cleats to leave their step treads in a horizontal position, the escalator allows its passengers enough time to establish themselves on a single step before the belt transitions to the inclined regime of steps. A typical escalator bank is shown in Figure 1.1. This particular installation is configured for two-way operation, with two parallel KONE escalators operating in each direction, separated by a bypass staircase.

Figure 1.1: Typical Escalator Bank, Dulles International Airport

With time, further developments in escalator technology have emerged, from handrails that move along with the steps to interlocking cleat designs that significantly reduce the risk of foot entrapment (Lampugnani, Hartwig, Simmen, & Imorde, 1994). However, it is interesting to note that throughout the last century escalator technology was developed much earlier than that of moving walkways. Although both types of moving belt systems are now comparable to one another, it is intriguing that the complicated, torque-intensive, and frankly dangerous system that makes up an escalator was developed in advance of the vastly simpler moving walkway. In light of the fact that it is only with the relatively recent advent of large indoor facilities that such a device is required, this lag begins to make sense. A typical moving walkway installation for two-way operation is shown in Figure 1.2.

Figure 1.2: Typical Moving Walkway Installation, Dulles International Airport

1.3 Design of Pedestrian Interface Areas

When designing public facilities, one aspect of importance to consider is that of pedestrian flow within the facility. Even the most attractive and functional structure will be unsuccessful if it proves to be too difficult for its users to traverse. Although a moving belt will not necessarily transform a structure that is dispersed horizontally or vertically into something with a tightly linked and effective floorplan, the inclusion of these systems can go a long way in bringing a spread-out structure closer together. Furthermore, even in a situation where a moving belt is not needed because of distance, their inclusion can help to promote rider comfort and convenience (O'Neill, 1974). Without the presence of an escalator, the area taken up by vertical transportation facilities would be prohibitively large, no matter whether the alternative transportation was provided by stairs or elevators.

To this end, architects and engineers must be sure to consider the dynamics of pedestrian circulation when developing these structures. Interface areas generally include facilities where there is a change of mode or vehicle, a characteristic perhaps best exemplified by a transportation station like a train station or airport. However, a shopping complex, office building, or other large public structure could also count in this category because of the fact that users must transition from the mode of their arrival to whatever activity may be occurring at the endpoint of their trip. The important behavior to note is the presence of pedestrian circulation within the facility, since this action must be accounted for in the design phase in order to assure that the final facility design will have sufficient pedestrian conveyance capacity.

1.3.1 **Necessary Simplifications**

Although the capability exists to simulate the movement of crowds throughout a facility, only a small percentage of interface areas are subjected to this level of analysis. Typically, simulation is only performed in facilities where crush loading conditions may occur that will create unsafe conditions for users. However, even in this situation, it is more common for designers to use architectural standards to determine the width of a corridor or the number of exit doors required based on the projected flow of people along that particular path.

With regards to escalators and moving belts, design is typically performed by determining the normal and peak loading of users who desire to travel along a particular pedestrian corridor. For instance, a designer may know that during rush hour there may be some number of people who desire to exit an underground subway station. Through the use of published architectural standards or projected loading curves, they can determine how many escalators will be needed to handle the outbound flow. Unfortunately, there are a number of flaws and imperfections that are present in this approach.

1.3.2 **Flaws in Belt Specification**

In the scenario described above, a designer wants to determine what configuration of escalators is necessary for subway installation. The current state of the practice would involve conducting research in a published manual or manufacturer's guidelines to determine the capacity of a given escalator configuration and then determining the quantity and specifications needed to meet demand. However, this method may prove to be unacceptably inaccurate under certain conditions.

1.3.2.1 *Determining Capacity*

At present, the capacity of an escalator or moving walkway is determined by looking at a manufacturer publication and finding the "practical capacity" based on the speed of operation permitted by that locality (ThyssenKrupp Elevator, 2004). Unfortunately, the methods behind this method are somewhat oversimplified.

To determine the practical capacity, the manufacturer first computes the theoretical transport capacity, which is equal to the number of treads per hour times the number of people each step can support. In conventional installations with a speed of 0.5 m/s, a tread depth of 0.4 m, and a tread with capable of holding two passengers, this value is 9000 passengers per hour:

$$theoretical\ capacity = (treads\ per\ hour) \times (passengers\ per\ tread)$$

$$[(belt\ speed) \times (tread\ depth)] \times (passengers\ per\ tread)$$

$$\left[(0.5\,{}^{m}/_{s}) \times \left(\frac{1\ tread}{0.4m}\right)\right] \times \frac{2\ passengers}{tread} = 2.5\,{}^{pass}/_{s} = \boxed{9000\,{}^{pass}/_{hr} = theoretical\ capacity}$$

Up to this point, the logic behind capacity determination for escalators is sound. However, in order to convert this theoretical capacity to practical capacity, the standard approach is to account for loading inefficiencies, thereby reducing the capacity down from this theoretical level. While this approach is backed up by observation and by research, the specific method by which the practical capacity is determined is through a multiplicative adjustment factor, usually 0.8 (ThyssenKrupp Elevator, 2004). By using what in most applications is a constant adjustment factor, the impacts of numerous complicating factors are effectively ignored, from loading delays to luggage to the added space required by walkers as opposed to standers. Although this problem is not unique to moving belt facilities as even highway capacity calculations contain the usage of constant adjustment factors, this does not mean that the current situation must be accepted.

1.3.2.2 *Accounting for Regional Differences in Pedestrian Streams*

While it is important to include the impacts of these complicating factors, it is equally important to realize that the desires and characteristics shown by users in different regions are not the same. That is, the speed and aggressiveness parameters that are seen in New York City will be different from those in Newport News, just as the behavior of a rider on an escalator in a transit station during peak hours will be more rushed than a passenger at a shopping mall will on a Tuesday afternoon. It is theorized that if pedestrian behavior could be changed within the context of the capacity model, the results would be applicable over a wider range of sensitivity scenarios.

1.3.2.3 *Effects of Crush Loading*

As mentioned above, simulation of pedestrian flows is sometimes used in situations where a sudden inflow of users can raise safety concerns, either through the potential for trampling or crushing or because of the risk of overflowing a limited loading area. An additional practical application of pedestrian flow simulation can be found in the study of emergency evacuation scenarios. To this end, it is suggested that microsimulation of moving walkways may similarly be able to account for the intricacies present under these conditions.

Under crush loading conditions, microsimulation would, for instance, allow the engineer to track the growth of a queue on a subway platform. If the queue reached some critical length, there may be potential for members of the crowd to fall onto the tracks, so the designer would ensure that this condition would not occur given projected train unloading rates. However, an additional benefit of the use of microsimulation in determining belt capacity would be that various parameters such as the ones discussed above might be implemented in the model.

1.4 Proposed Approach to Belt Capacity Analysis

Because of the perceived drawbacks present in the current state of the practice in moving belt capacity analysis, this paper proposes the development of a microsimulation framework that describes and quantifies pedestrian flow on moving belt surfaces.

1.4.1 **Model Operation**

This model would be capable of taking in a range of relevant input parameters, from pedestrian desires and choices to belt specifications and operational characteristics. This information would be used to define the simulation environment, at which point a logical framework and rules would be used to govern the interaction of all participants within the system. In this way, all automata within the model will exhibit behaviors based on their own desires within the context of the overall system rules. Through the interaction of every entity, the system-wide characteristics may be tabulated over time to determine relevant operational measures of effectiveness.

1.4.2 **Source Data**

In order to adequately define the rules that govern each entity's behaviors, a literature review will be undertaken in order to quantify and codify the choices made by a person in an activity as mundane as walking or stair climbing. As will be seen below, because of the relative lack of research conducted on the specific topics of following and passing behavior while climbing a moving belt, this information will be pulled from a diverse array of fields. While pedestrian behaviors like stair climbing and walking speeds can be taken directly from applicable studies, additional information will be taken from research conducted on walkers within bottlenecks and applied to the constrictions that are present in an escalator.

1.4.2.1 *Inclusion of Automotive Choice Behavior*

Furthermore, to account for the thought process that results in following behaviors and passing choice, information originally developed to model highway driving will be integrated into the model. Although it may seem improper to use automotive following and passing rules to determine the behavior of pedestrians, the means by which these human factors decisions are made are actually very similar by virtue of the fact that in both cases it is a person who is making the choice. Additionally, since these automotive rules are relatively well funded, there exists a much larger and more substantial basis of work on which to base this section of the model.

1.4.2.2 *Sensitivity Analysis Capabilities*

Through the use of research from a variety of unconventional sources in a novel approach to moving belt analysis, these unaddressed factors may be included in the transportation analysis and decision-making process. Furthermore, it will be possible to vary the input factors slightly to determine how sensitive the solution is to slight changes in any number of characteristics, from input stream volumes to belt speed to traveler characteristics. In this way, projected changes in passenger mix or operational rules can be investigated and accounted for in addition to increases in passenger volumes.

Chapter 2 – Literature Review

The state of the practice of designing moving belt facilities has several key shortcomings, as were discussed in the previous section. In order to rectify these issues, this report proposes a new microsimulation framework of pedestrian behavior to gain a clearer understanding of the actual capacity of a belt system. This model will include key elements of pedestrian behavior while at the same time accounting for the variabilities in pedestrian mix and aggressiveness as well as any operational rules or restrictions that may be in place on the facility.

However, because the literature indicates no projects that have attempted this approach, at least not with the same conditions and constraints that are present on a moving surface operating in a narrow bottleneck, the theoretical framework that drives the model had to be developed from the ground up. In order to accurately model the behaviors present within this sort of system, information needed to be pulled from a number of sources, from pedestrian behavior to automotive traffic flow theory. Additional references covering the operation of a belt system as well as the practice of determining capacity and level of service in such a system have been pulled as well. Finally, an investigation of varying modeling practices was also conducted to find a suitable means of creating the desired simulation.

2.1 Pedestrian Behavior

The foundation of the model must come from the basic behaviors exhibited by pedestrians using the facility. Significant literature exists describing the capabilities of pedestrians in terms of their travel characteristics and how people tend to use the space available within a system. From this base, the model can be strengthened by the inclusion of various complicating factors, as these constraints will cause the entities present in the model to be restrained as they would in the real world by impediments like stairs and entry bottlenecks.

2.1.1 **Unrestricted Condition**

At the most fundamental level, pedestrian behavior is defined by how a person moves in open space. There are a number of characteristics that pedestrians exhibit both while moving and in terms of how they fill space.

2.1.1.1 *Travel Characteristics*

Escalators and moving walkways have been designed to fit the size and physical capabilities of locomotion that humans exhibit. Therefore, this model must be based on these sizes. At rest, the average adult male can be approximated by an ellipse with dimensions of 18” by 24” (0.45m by 0.6m). It is important to note that this design ellipse is a conservative size selected to account for a number of factors, including larger bodies, luggage and other personal articles, and body sway while walking (Lee, 2005).

At the same time, the average person is capable of sustaining a maximum walking pace of 240 feet per minute (O'Neill, 1974), or about 1.2 meters per second. Any pace above this speed would involve both feet leaving the ground simultaneously and would best be classified as running. In maintaining this level of effort, the average stride length is 1.58m for males, compared to 1.32m for females (Sutherland, Olshen, Biden, & Wyatt, 1988). It is important to note that this measurement is given in terms of stride length and is therefore measured as the distance between steps of the same foot. The distance between each individual step is known as the step length and is equal to one-half of the stride length.

As will be seen later, belt systems have been made in such a way that these parameters are within the physical capacity of the system. In this way, escalator steps are deep enough (0.4 meters deep) for riders (on average 0.3 meters deep) to occupy consecutive steps, just as two riders can stand side by side on a moving walkway. Similarly, the operation of belts is restrained by a number of factors including safety and the ability of humans to handle the interface between a moving surface and a stationary one. Because of these limits, the speed at which a belt operates (no greater than 180 feet per minute) is held below the maximum unassisted walking speed (240 feet per minute), thereby allowing users to overcome the motion of the belt and avoid a potential hazard if necessary.

2.1.1.2 *Available Space*

According to the literature, there are four basic elements involved in the dynamics of crowd motion: time, space, energy, and information. Time covers the way in which pedestrian flow peaks and ebbs across a period of time, space deals with how the crowd utilizes the available area, energy indicates the amount of effort exerted by the individuals and how this information travels through the crowd in the form of shockwaves, and information means the way the crowd becomes aware of instructions or hazards (Fruin, 1984). In this last case, information can be transmitted through formal means like a public address system or a dynamic message board or informally through conversation or rumor, the latter of which can cause a significant safety threat if the crowd begins to converge on a single constricted point.

Although all elements of crowd dynamics are important in the context of designing an ambulatory facility, the factor that impacts pedestrian behavior most on the belt itself is that of space. At the most basic level, space utilization is governed by size of a person and any belongings they may happen to have with them. However, another important restriction is formed by the social conventions that exist which drive people to avoid body contact with others (Lee, 2005).

As before, the size of a human body can be represented by an ellipse. At a bare minimum, the body takes up an ellipse with an area of 1.5 square feet (Fruin, 1984). However, this size does not include the spacing for sway and other factors that was outlined by Lee. Rather, this area allows a person no control over their own movement, as with this level of spacing they are in direct contact with others. Fruin states that "at approximately 3 square feet per person, involuntary touching and brushing against others will occur, a psychological threshold that should generally be avoided in most public situations. Below 2 square feet per person, potentially dangerous crowd forces and psychological stresses may begin to develop" (Fruin, 1984).

In this condition, shockwave propagation will be direct and unrestricted, leading to danger if a crowd in this condition is rigidly confined by architectural features like corridor walls, doorways, guardrails, or even escalators. An additional danger comes from the mechanical conveyance that is inherently present in escalators only serves to reduce the individual's control over their own movement as well (Fruin, 1984).

In order for pedestrian movement and choice to occur, density must be reduced. As occupancy decreases, the amount of space available per person will see a corresponding increase. At a level of 5 square feet per person, pedestrians are able to stand stationary without touching each other. Fruin suggests that this level is the minimum "in most normal waiting situations," listing several examples including standing onboard an escalator. At 10 square feet per person, walking becomes possible on the part of the pedestrians, provided this movement is done in cooperation with others. Finally, at 20 square feet per person free movement is possible (Fruin, 1984).

However, in addition to the body ellipse and the buffer zone that surrounds it, another important factor of spacing that should be considered is the "pacing zone", that is, the area required for stepping forward while walking (Lee, 2005). It is this level of spacing that raises the amount of area required per person to high levels in the Fruin estimates. However, still further space is required to detect and respond to the terrain and conditions ahead, which Lee calls the "sensory zone". Although this area does not need to be kept entirely free of other pedestrians for the user to make sound decisions, it needs to be of relatively low density for them to satisfactorily respond to obstacles ahead.

2.1.2 **Complicating Factors**

The speed at which a pedestrian is capable of moving is determined based on how they perceive the surrounding environment. The information collected in this scan of the immediate area is coupled with that individual's personal characteristics as well as the characteristics of the trip they are making – that is, the trip's purpose, the presence of luggage, and their familiarity with the route they are taking. The infrastructure present, including the level of elevation change and presence of shelter or climate control has a factor in their speed choice, as do environmental factors like the weather or even the attractiveness of the environment in terms of the presence of shops or the possibility of crime (Hoogendoorn & Daamen, 2005).

When applied to the case of a moving belt system, several of these elements are of concern in the context of a modeling situation. In addition to the obvious impedement caused by steps or elevation change, users will also experience delay from factors like bottleneck effects or the presence of other riders. These factors will serve to hinder the baseline, unrestricted walking parameters described above.

2.1.2.1 *Stair Climbing*

The speed at which a person is capable of climbing or descending a set of stairs will almost invariably be less than their speed if they were traveling along level ground. A study conducted by Fujiyama and Tyler was designed to determine the average climbing speed of different classes of pedestrians, grouped by age. They found horizontal speeds of 0.44 to 0.76 m/s while ascending and 0.47 to 0.87 m/s while descending. While this range is fairly wide, the variability could not be tied to any variable other than the preferences of individual users (Fujiyama & Tyler, 2004)

Additionally, the study found that there is not a significant difference in walking speed between the old and young classes, nor is there a significant difference present when the user is climbing at a "normal speed". However, when asked to ascend and descend the stairs quickly, a difference emerged at the 95% significance level or above. The researchers concluded that above a certain point, leg strength does not benefit the walking task. Younger users may be capable of traveling faster if desired, but a "normal" climbing speed does not reach the maximum speed of elderly persons and therefore the average speed for all users under normal conditions is similar regardless of age or gender (Fujiyama & Tyler, 2004).

Other studies have found largely similar speeds when adjusted to give the horizontal, unassisted pedestrian speed. A study of the London Underground gave a value of 1.40 m/s for walkers onboard descending escalators across all user classes, which when the escalator speed is factored out and the diagonal speed is converted to a horizontal one gives a speed of 0.55 m/s (Davis & Dutta, 2002). Similarly, a study on the Washington Metro shows a stair-climbing speed of 0.8 m/s, which corresponds to a speed of 0.69 m/s (Sutthawassuntorn, 2010). Using this range of values, it is possible to determine a range of walking speeds on stairs to go along with the previously defined spread of unrestrained walking speeds.

Another important factor to consider when analyzing stair climbing motion is the fact that travel along a staircase is restricted by the constraints of the steps themselves. That is, the dimensions of the stair riser and tread surfaces will limit the number of possible locations for a user to place their feet while ascending or descending. Therefore, the user will find themselves limited a pace where the average step length equals the depth of the stair tread (Lee, 2005).

2.1.2.2 *Elevation Change*

It is not just the presence of stairs that creates an impediment to pedestrian motion; the elevation change itself can prove a detriment to walkers. First, impedance is observed as a climber gains elevation through the strain of physical exertion. The maximum speed a pedestrian desires is seen to decrease slightly as their climb progresses. However, another important detriment to forward progress comes from the impact that elevation change can have on sight lines.

Various studies have shown an interesting relationship between the jam density observed on a staircase and the direction of travel along that facility. Specifically, the spacing between users is observed to increase when travel is in the uphill direction. This behavior was determined to be a result of a "facial obscuring". Facial obscuring occurs when there is a lack of forward space available in front of a user, and specifically in front of their face. When traveling in the uphill direction, a user standing directly in front of the rider in question will – by virtue of their higher standing position – block all of the rider's view since the rider will be staring directly at their shoulders or back. This violation of the forward facial ellipse can cause hesitation or discomfort on the part of the rider and causes greater open space to exist when traveling uphill. In the downhill direction, higher densities are observed because the opposite effect occurs: users in front of the rider in question are standing on a lower step, and therefore the rider can see over the top of the forward user's head, giving the illusion of greater forward space (Lee, 2005).

This facial ellipse concept is illustrated in Figure 2.1, below. It is interesting to note that this figure has been created through a modification of the standard copyright-free United States Department of Transportation escalator pictogram, as developed by the former American Institute of Graphic Arts (Geismar, Chwast, de Harak, Lees, & Vignelli, 2011). In this figure, the standard solitary escalator rider has been copied and the facial ellipse highlighted in yellow to show the conflict that arises when traveling in the upwards direction.

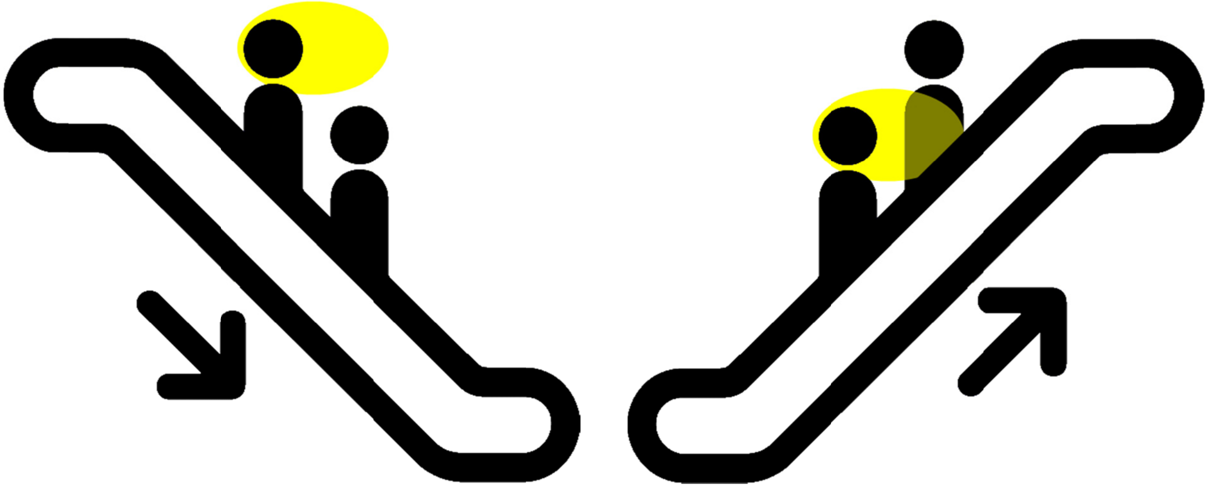


Figure 2.1: Facial Ellipse Effect in the Downwards (Left) and Upwards (Right) Directions

2.1.2.3 *Bottleneck Effects*

Another feature of moving belt facilities that must be accounted for in the proposed model is the constricting effect that the balustrades – the railing structures that are required by code on all contemporary escalators and moving walkways – have on the pedestrian stream. An escalator and its glass balustrades are shown as Figure 2.2, showing the constricting, corridor-like effect that is caused by the handrails.

Fortunately, there exists a sizeable body of research on flow through bottlenecks. Most notably, Serge Hoogendoorn runs a lab at the Delft University of Technology where volunteers are sent through experimentally designed constrictions to produce microscopic pedestrian flow data (Daamen & Hoogendoorn, 2003).

Figure 2.2: Escalator with Glass Handrail Balustrades, Dulles International Airport

Insights gained from these practical experiments indicate that while shoulder width alone indicates that it may be possible for two pedestrians to operate side by side in a narrow bottleneck, the lateral space that is required as pedestrians attempt to maintain separation from the side walls as well as each other prevents this from occurring. Because of the fact that pedestrians within the bottleneck both want to maintain their forward sight lines and achieve a satisfactory level of horizontal spacing, a behavior that Hoogendoorn calls "zipping" emerges. Zipping patterns are characterized by each subsequent user taking up a position to one side or the other within the bottleneck, much like the teeth of a zipper. In this way, the strides of consecutive users do not interfere with each other, an individual pedestrian is able to identify forward obstacles, and lateral spacing is maintained (Hoogendoorn S. , 2004).

In a following experiment, it was determined that the overlap present between the parallel, zipped layers of pedestrians links the flow of each stream together, despite the fact that they are in different "lanes". Because a pedestrian in one layer takes up more than half of the corridor, it becomes difficult for another user in the adjacent layer to pass, thereby making the speed of each stream generally constant. Additionally, the shoulder sway exhibited by a pedestrian as they walk makes passing nearly impossible. To this end, the study concludes that the capacity of a bottleneck is stepwise, based on the width of the corridor and how many layers or lanes it can support (Hoogendoorn & Daamen, 2005).

Since Hoogendoorn's experiments generally take place in a meter-wide bottleneck – the same width as a standard escalator or moving walkway – it is believed that his findings can be applied to the proposed model. However, the experimental pedestrians desire to maintain separation from the walls as well as each other, unlike the stationary pedestrians on an escalator who can choose to "hug" the railing, which allows other users to pass in a parallel layer. This behavior is supported by the inclusion of physical walk/stand lane markings found in ambulatory facilities throughout the United States, as seen in Figure 1.2 and especially in Figure 2.3, below.

Figure 2.3: Moving Walkway with Lane Markings, Minneapolis-St. Paul International Airport

A subsequent study detailing the difference between single-file flow in a narrow corridor and the flow of pedestrians in a wider corridor confirms the "zip effect" observed by Hoogendoorn. However, this study goes one step further in that it observed this behavior on escalators in the London Underground. Additionally, that report found the presence of a "capacity drop" that occurs when a constrained corridor system like an escalator goes from normal flow to saturated flow, as this transition causes the packing structure of the pedestrians to adjust to handle the higher inflow. It is the transition itself that causes a temporary but significant drop in flow, the presence of which can significantly disrupt the operation of a system. Because of this drop, the study proposes metering users in such a way that the peak inflow does not exceed this maximum sustainable flow rather than the capacity of the system since this would incur the capacity drop (Cepolina & Tyler, 2005).

Within the context of a transit system, the function of metering could be approximated through progressive platform design. Specifically, by positioning the escalator in a way that prevents all those users disembarking a train from reaching the escalator at the same point, the peak inflow is spread out to keep the escalator operating in a sustainable manner.

2.1.2.4 *Following Behavior*

Again, Hoogendoorn's research provides an answer to another critical impediment to pedestrian motion: the presence of other users within the bottleneck. Using experimentally determined pedestrian headway observations, a composite headway distribution model was developed. This model is called a composite model because it contains two sets of behavior equations, one for those who are unconstrained and another for those who are constrained by the fact that they must follow another user. Above some free headway, the model states, it can be said that a pedestrian is not constrained by any other pedestrians present ahead. This desired empty zone is unique to each pedestrian and assumed to be exponentially distributed as a result of user choice as well as individual variations in step size and frequency (Hoogendoorn & Daamen, 2005).

However, the equations contained within the composite headway distribution model require calibration based on field data to fully operate the model. Furthermore, the data provides a distribution of headways for the overall system, which would be difficult to implement in a microsimulation-based model like the one proposed in this paper. Additionally, basing each simulated user's actions on the goal of maintaining a specific headway is believed to not model real-world conditions as well as a model in which the user varies their speed directly based on their observed headway.

2.1.2.5 *Passing Choice*

There does not exist much information about how a walker decides to pass another in free space. Blue and Adler present a logical method for determining when a pedestrian desires to change lanes to walk around a slower user, but this method is based on passing in open space (Blue & Adler, 1998). Therefore, this technique is unsuitable the purpose of modeling pedestrian behavior in a constrained system like what exists in a bottleneck or corridor.

In a corridor scenario, pedestrian behavior tends to form informal but rigidly defined lanes similar to those seen on a highway. While passing behavior may not be formally codified, many moving belt installations operating in public transit systems show an informal trend where those users who choose to stand on the belt stand to the outside lane, leaving the inside lane available to those who choose to walk. In fact, Davis and Dutta report that the London Underground is perhaps unique in that the transit authority has installed signage in several stations instructing users to keep to the left – the outside lane in Commonwealth nations – unless actively passing someone. In this way, both standers and slower walkers are kept out of the passing lane unless occupancy reaches the point where both lanes must be filled (Davis & Dutta, 2002). Because of the similarities that exist between passing behavior on a highway and passing behavior of pedestrians within a constricting corridor, this project will look towards developments that have been made in traffic flow theory in order to model these parameters.

However, further understanding can still be gained within the field of pedestrian passing. A subsequent paper by the Blue and Adler proposes that peak of lane changes will occur where an occupancy of between 20 and 40% is observed. This occupancy is defined as the number of "cells" occupied, where each escalator or moving walkway tread with depth of 0.4 meters is split into two standing positions, yielding a cell size of 0.4 meters deep by 0.5 meters wide. Blue and Adler report that these occupancies "are presumed to be the speed-flow-density combinations that have the most volatile dynamics", meaning that this range is where the best gains may be had from a lane change (Blue & Adler, 2001). Below this density there will be enough free space to allow users to travel unimpeded, thereby keeping lane changes to a minimum. Conversely, above this density the efficiency of a lane change is greatly reduced because acceptable gaps will be hard to find in the adjacent lane.

While the formation of lanes has the potential to result in a loss of capacity because of the inefficiency that comes from suboptimal packing of riders on the belt (Daamen & Hoogendoorn, 2003), the tendency of users to form lanes provides several benefits. In most non-saturated conditions, the outside lane will exhibit dense standing, either every step or on alternate steps depending on whether the belt is traveling uphill or downhill, while the inside lane will carry walkers traveling at a higher effective speed (Sutthawassuntorn, 2010). Because of this operation, the benefits of lane formation may be realized in terms of user satisfaction, increased speeds in lower flow periods for those who desire to walk quickly, and also in that it provides order in an otherwise chaotic system. To address the potential for capacity to be increased by prohibiting passing behavior, the proposed model will incorporate a number of rules that govern belt operation so that various rule-based scenarios may be tested.

2.2 Traffic Flow Theory

Because of the similarities between pedestrian flow within a corridor and the operation of vehicles along a roadway, techniques related to traffic flow theory were reviewed to form the basis of movement within the model. Although Hoogendoorn and Bovy state that "vehicular flow simulation modeling approaches are generally not applicable to pedestrian flow modeling" (Hoogendoorn & Bovy, 2003), they fail to provide justification of this assertion. For this project, it is strongly believed that the constraints present on a moving belt surface through the bottleneck-like balustrades and the social convention of lane formation make moving belt systems very much analogous to automotive behaviors, as will be seen below.

2.2.1 **Following Behavior**

Because of the level of attention that has been paid to traffic flow theory over the past few decades, there exists a sizeable body of work including numerous mathematical approaches to the situation of car-following behavior. Each of these models considers slightly different variables and utilizes a different weighting scheme in an attempt to turn measurable parameters about the vehicle stream into a predictive framework for following behavior.

2.2.1.1 *Existing Derived Equation Models*

A number of these models can be found in the literature. One of the first vehicular car-following models to be developed was created by engineers from the General Motors Corporation around fifty years ago. The GM models are a set of stimulus-response models of car-following behavior, where the stimulus comes from changes in the headway present between the two vehicles in conjunction of the speed of the lead vehicle, while the response of the model is shown in the resulting acceleration or deceleration of the following vehicle. Thus, the extent of the response of the following vehicle is directly related to the speed of the lead vehicle and inversely related to the headway present between the two, implying that as spacing increases the actions of the lead vehicle have a diminished effect on the speed of the following one (May, 1990).

Since the development of the GM models, several other models have come along. Next came the Greenshields model which attempted to relate macroscopic traffic characteristics to the speed exhibited by a following vehicle (Rakha & Crowther, 2002). Subsequent models like Gipps (Rakha, Pecker, & Cybis, 2007), Pipes, and Van Aerde (Rakha & Crowther, 2002) have all become more advanced, accounting for various other potentially useful parameters in an attempt to accurately model the decisions a driver makes as a result of the raw data inputs that they could potentially receive.

2.2.1.2 *Rule-Based Models*

However, within the context of the pedestrian analogue, it is perhaps more desirable to have following behavior governed by a rule-based method instead of a complicated series of equations. While car-following behavior is easily broken down to its operational components, in that a driver can express their desired speed through the analog input of the accelerator pedal, pedestrian following is a far more organic and innate ability. Humans are hard-wired to leave some amount of spacing and maintain a constant headway to those around them without really having to give it much thought, and in order to program these goals and desires into a complete modeling framework they must be distilled into a straightforward yet comprehensive rule set.

To this end, a rule-based model of car-following was found in the literature as a component of the TRansportation ANalysis SIMulation System, or TRANSIMS. TRANSIMS bills itself as "the next generation planning/simulation model" because of the way it implements advances in computing technology and data availability to perform travel demand analysis and forecasting at the network level while modeling pedestrian, automobile, and transit operations at the individual level through microsimulation (Hobeika & Gu, 2004). In this way, TRANSIMS gives all the benefits of the conventional four-step model of transportation planning while at the same time accounting for micro-level behaviors like route planning and regional congestion.

The benefit of TRANSIMS to the creation of the proposed model comes from its microsimulator module. Because TRANSIMS is designed to handle large networks, the developers of the system wanted to ensure that the software would not be bogged down by complex calculations of desired acceleration or other factors as is used in the equation-based models described above. In order to combat this issue, they decided to implement a coarse simulation approach using a cell-based network, described below, along with rule-based behavior algorithms (Hobeika & Gu, 2004).

The resulting algorithms are based on two basic rules. First, a user wants to accelerate wherever possible, up to their desired maximum speed. Second, deceleration will occur only if necessary, albeit with some low probability of random deceleration to mimic real-world behaviors. Because of the simplicity and elegance of these rules, the only input variables required in the microsimulator are the present speed of the subject vehicle, its maximum attainable speed – whether limited by the vehicle or by a speed limit, and the forward headway (Los Alamos National Laboratory, 2006).

Using these parameters, at each time step every vehicle on the network computes the forward gap ahead and makes decisions based on the relationship between its current speed and the forward gap. If the speed is greater than the gap, it slows down to a speed ensuring that it will not violate the gap in the next time step no matter at what speed the lead vehicle is traveling. Otherwise, and if the current speed is less than the maximum speed, it will accelerate by whatever acceleration factor the vehicle is capable of achieving. In each resulting case, there is some low probability – usually 5% - that the vehicle will slow by some increment for no particular reason other than to mimic distraction and other actual behaviors, while at the same time giving the system a bit of randomness to keep it from settling into an unrealistic situation (Los Alamos National Laboratory, 2006).

2.2.2 **Passing Choice**

When it comes to modeling passing choice, human behavior is based on just that – choice. Lane change behavior is a discrete, binary decision process based on logic and rules rather than some continuous numerical calculation. Therefore, a rule-based approach to lane changes and passing is the only practical approach to use in the proposed model, and because of the clearly defined analogy between lanes on a highway and the lane-like layers observed in pedestrian flow within bottlenecks it seems that traffic flow theory is once again uniquely suited to this particular problem area. Fortunately, TRANSIMS was created with a set of rules to govern passing behavior in addition to the previously discussed following algorithm.

The TRANSIMS microsimulator contains two different sets of lane change algorithms, one to govern the passing of slow-moving vehicles and one to ensure that a vehicle is able to be in the appropriate lane to make a required turn along their route. The proposed model will utilize the first algorithm, which contains several conditions that must all be met for a lane change to occur (Los Alamos National Laboratory, 2006).

The first condition that must be met is that the gap forward in the current lane is less than the distance the vehicle would travel in the next time step. If this proves true, the vehicle begins scanning the adjacent lane with the second condition, that the gap forward in that lane is greater than the gap forward in the current lane. There is no sense changing lanes if the forward gap there is less than what exists at present. Next, the microsimulator checks to determine if adjacent lane's gap is greater than or equal to the current speed of the vehicle. If and only if all of these criteria are met, the model makes a final check to ensure that any lane change that occurs will not cut off a vehicle behind in the adjacent lane by checking that the available gap backwards in the adjacent lane is at least as large as the maximum speed that any vehicle on the network could be traveling. This is to ensure that no matter what the capabilities of that adjacent back vehicle, they will not be traveling so fast as to collide with the subject vehicle at the next time step (Los Alamos National Laboratory, 2006).

In addition to the TRANSIMS-defined procedures for changing lanes, the proposed model will also include additional decision logic that will serve to drive slower users back into the outside lane when faced with a faster walker approaching from behind. In this way, the users will keep right whenever possible in order to ensure that the passing lane is kept free from impediments.

2.3 Belt Characteristics

Now that sufficient background exists to define the behavior of the pedestrians within a moving belt system, the system itself needs to be constructed. Fortunately, the public safety concerns that impact the operation of escalators and moving walkways means that there exists a sizeable amount of documentation and regulation that applies to the design and capabilities of these facilities.

2.3.1 **Geometric Specifications**

First off, the physical layout of a moving belt system is standardized to facilitate ease of use, ergonomics, and public safety. The individual tread dimensions are mandated such that only specific step widths of 24”, 32”, and 40” (up to 1.0 meters) are allowed, with significant restrictions present on the situations where narrower treads may be used. The rise and run of each step are constrained to be 7-7/8" (0.2m) and 15-3/4" (0.4m), respectively, according to a manufacturer (Schindler Escalators, 2010). These specifications are confirmed by American Society of Mechanical Engineers’ *Safety Code for Elevators and Escalators* (Donoghue, 1981).

At present, the most common tread width on escalators and moving walkways is 40 inches, or approximately one meter. The reason for the prevalence of this particular tread width is that it is the minimum width that can adequately support two independent travel lanes for pedestrians. The reason that wider belts are almost never seen in practice is that a wider belt would encourage the formation of a third lane in the center of the belt. This behavior could prove to be dangerous, as those pedestrians traveling in the center lane would not be able to reach a handrail during saturated conditions, so belt widths are kept to one meter to prevent this third layer of pedestrians from forming (O'Neill, 1974).

Additional specifications also exist that are not as pertinent to the operation of the model. One such specification states that the requirement that three flat steps exist on each end on low-rise escalators of less than 10 meters with at least four steps mandated on taller installations. Another specification mandates that the level of inclination of an escalator belt is not to exceed 30 degrees (Welch, et al., 2009). This parameter was used to convert diagonal distances and speeds into horizontal ones.

Other sources provide alternate values for the specified angle of inclination. ASME requires that escalators not exceed 30 degrees with an allowable margin of error of plus or minus 1 degree, meaning that contractors can technically “cheat” upwards in steepness as long as they are confident of their installation precision (Donoghue, 1981). European specifications allow an incline of 30 degrees but permit installations up to 35 degrees in the case of low belt speeds (Bangash & Bangash, 2007).

For moving walkway installations, the commonly accepted requirement is that the incline of the belt should not exceed 12 percent (Bangash & Bangash, 2007).

A diagram illustrating these specifications is shown as Figure 2.4. This figure has been created by annotating a picture of a KONE escalator found at Dulles International Airport. Note the yellow and black markings, which are suggested by the ASME *Safety Code* to highlight the edges of each tread for safety (Donoghue, 1981).

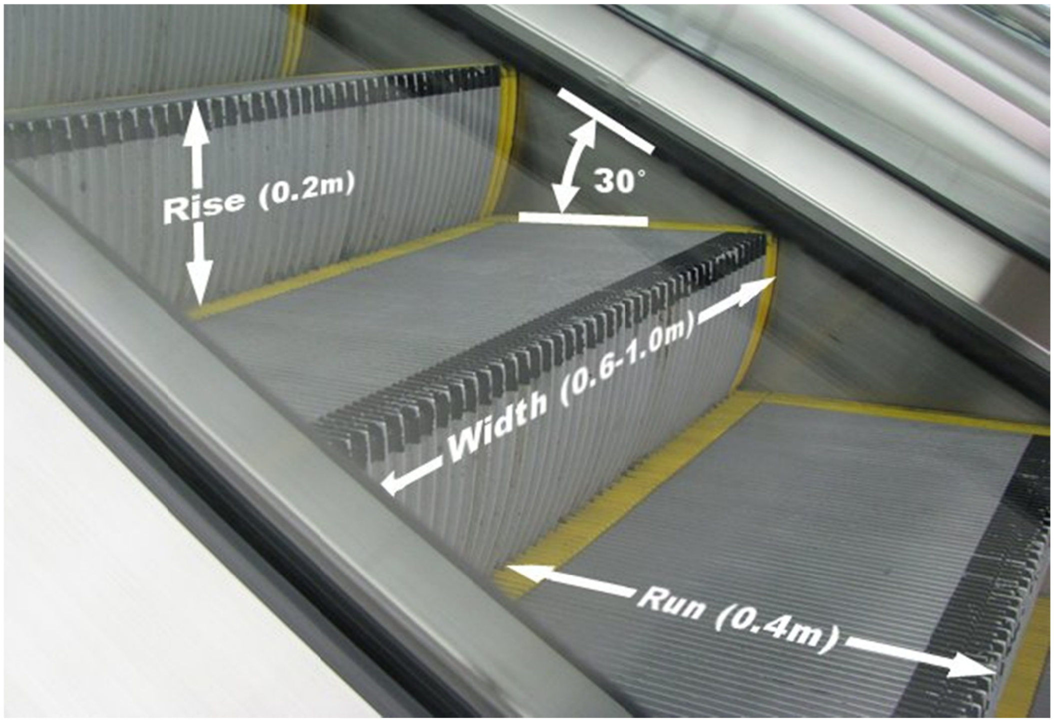


Figure 2.4: Standard Escalator Dimensions

2.3.2 **Operational Parameters**

An additional area of concern for governmental regulators is the speed at which escalators and moving walkways are allowed to operate. The most typical requirement on the part of municipal codes is that the speed of an escalator should not exceed 90-100 feet per minute, or 0.44 to 0.51 meters per second (Welch, et al., 2009). However, the ASME code provides the opportunity for localities to approve operation at a higher speed provided that it is deemed safe by local regulatory officials, although this provision is rarely taken advantage of (Donoghue, 1981). Additionally, various transit agencies such as the London Underground have conducted studies with escalator belt speeds of up to 180 feet per minute, or 0.91 meters per second (O'Neill, 1974), although this is well in excess of the typical operation speed within that system of 0.72 meters per second. Generally, moving walkways are permitted to operate up to 180 feet per minute, so long as their degree of inclination is kept low (Donoghue, 1981).

In the O'Neill study that described London's experiments, an assessment of the throughput possible under varying operation speeds was also included. It was determined that an operational speed of 145 feet per minute (0.74 meters per second) would provide the maximum flow of passengers through a moving belt system, whether an escalator or a moving walkway

(O'Neill, 1974). The stated reason for this occurrence deals with the interface between the belt and the stationary floor surface on either end. While a high belt speed allows for faster speed of movement, higher speeds cause trouble for users when they have to board or disembark from the belt. Not only is it hazardous to change from a constant platform to a moving one and vice versa, concerns are also raised by the potential for dangerous pileups to occur if a moving belt continues to convey passengers into a crowed area.

2.4 Capacity Analysis

As was previously described in Section 1.3.2.1, determining the capacity of a moving belt system is at present a very approximate endeavor. By combining an idealized occupancy with the belt speed and then adjusting by some arbitrary occupancy factor, the present state of the practice fails to account for a number of key factors while developing a value that is not sensitive to changes in pedestrian stream characteristics.

2.4.1 Empirical Capacities

Measured flow rates for escalators generally give capacities in the vicinity of 4000-6000 passengers per hour. This range of values was found across a variety of sources, including 5400 from the manufacturer publication shown previously (ThyssenKrupp Elevator, 2004), a slightly higher estimate of 6400 passengers per hour from Fruin based on occupancy statistics alone (Goodman, 1992), and an empirically defined range of about 4100 to 5400 passengers per hour based on the combination of tread width, belt speed, and incline present on that particular unit (Turner, 1998). O’Neill further confirms these values, albeit on a 48” escalator which is a size no longer permitted by legal codes (O'Neill, 1974). Subsequent studies have gone further in analyzing the capacity of each layer of an escalator to give the capacity of both the standing lane and the walking lane under a representative scenario of choice behavior (Davis & Dutta, 2002) (Sutthawassuntorn, 2010).

2.4.2 Level of Service Determination

As with most transportation facilities, the level of service of a moving belt system can be determined by computing its volume-to-capacity ratio and occupancy rate. These factors allow the engineer or planner to determine what amount of the facility’s capacity is being utilized. For stairs, escalators, and moving walkways this approach is the only real way of measuring service quality since in all but the most extreme peak loading conditions delay is not an applicable service metric.

In terms of volume-to-capacity (v/c) ratio, one analysis performed on the Washington Metro was developed such that stations which possessed one or more facilities with a v/c ratio at or above 0.75 were deemed to need enhancement. A v/c ratio between 0.5 and 0.75 flagged that station as needing a more detailed study of its pedestrian facilities (WMATA, Access and Capacity Study, 2008).

To give a more descriptive value of service quality of its pedestrian facilities, Washington Metro published a guide at the same time as the previous study that detailed the methods by which level-of-service grades could be assigned to these facilities. The determinations were based on a desired space per pedestrian – in effect, the inverse of occupancy – as well as the sustainable flow per unit width on the facility. The ratings ranged from LOS A, where a sufficient area per pedestrian exists freely select speed and to pass slower-moving users all the way to LOS F where "complete breakdown in pedestrian flow with any stoppages" is the norm. Using these criteria, the acceptable service level for pedestrians on stairs and moving belt facilities was deemed to be LOS C, defined as a space per pedestrian of 10 to 15 square feet. While speeds are observed to be slightly restricted due to the inability of fast walkers to pass other slower-moving pedestrians, this behavior is possible within the norms of gap acceptance in the adjacent lane. At LOS C, standing riders on belt systems start to cause noticeable conflicts (WMATA, Site Planning Manual, 2008).

2.5 Modeling Framework

With the collection of information regarding pedestrian behavior and dynamics compiled, it became time to determine how best to construct the simulation. The goal in selecting a simulation approach and software was to find an effective framework that would provide the desired simulation capabilities. First, the simulation needed to have the ability to create, alter, and track the current and desired states of a multitude of individual entities as they move through the simulation space. Once the entities were inserted into the simulation space, a predefined set of rules and behavioral parameters would govern the interaction and progression of all entities. This simulation space itself was another key capability, in that the selected model framework would need to possess the ability to display the current condition of the model at any instant in time to ensure its operation and adherence to real-world operations.

2.5.1 Simulation Approach

According to Hoogendoorn and Bovy, there are three model types that can be used to describe flow-based systems like those found in transportation engineering problems. At the broadest level, macroscopic modeling uses system-wide parameters like average flow and density to describe the condition of the system. Narrowing in somewhat, mesoscopic models like those found in gas flow models do not distinguish between the individual players in a system, instead choosing to use velocity distributions to simulate their behavior; but they still provide for interaction between the entities in the system through rule-based approaches. At the most detailed level are microsimulations like the one sought for use in this project that seek to describe the behavior of an entire system through the choices and interactions of those entities it contains (Hoogendoorn & Bovy, 2000).

2.5.1.1 *Cellular Automata (CA) Framework*

In the course of investigating TRANSIMS and its rules of driver interaction, a similarity was noted between the approach utilized by the TRANSIMS microsimulator and the constraints imposed by the tread spacing on an escalator. TRANSIMS uses a coarse simulation methodology known as cellular automata, or CA. CA divides each link found on the network into a number of regularly spaced cells, each of which is the size of one system user. Each user is progressed through the network using a simple set of rules, with the qualification that each movement must be some integer multiple of the cell size such that the occupancy of each cell remains one (Hobeika & Gu, 2004).

2.5.1.2 *CA Approach to Pedestrian Flow Modeling*

Although cellular automata simulations are generally more common in other fields like computer science or automotive behavior, they have seen some implementation in pedestrian simulations.

Blue and Adler constructed a series of models that described pedestrian flow a through one-dimensional free space using CA principles (Blue & Adler, 1998) (Blue & Adler, 2001). These models contained rules that governed the interactions between pedestrians, however, those approaches were deemed to be inferior to the previously defined TRANSIMS rule sets.

However, the Blue and Adler experiments provide some invaluable guidance in applying CA characteristics to the pedestrian paradigm. Most notably, the continuous speed distributions observed in pedestrian behaviors must be reduced to a series of discrete speeds, all in increments of one cell per second. Fortunately, since the configuration of an escalator limits the locations where it is possible for a user to place their feet to the tread surfaces, it was decided to make the cell size in the produced model equal to the depth of a tread, 0.4 meters.

After a range of walking speeds was constructed using this setup, a problem was discovered. Using a full tread as the cell size does not allow enough variation in the possible walking speeds to provide suitable sensitivity in walking speed. Consequently, it was decided to adjust the speed choice options to half-cell increments per second, giving twice as many options for the walking speed of the system users. As it turns out, this change allows for the inclusion of a previously omitted characteristic of human behavior, step straddling. This practice occurs when a user has a foot on one step and their other foot on an adjacent step, effectively causing the user to occupy a position midway between two steps. With the inclusion of following behavior, spacing between users will be maintained even in the face of a rider straddling steps, with the added benefit of additional positions present in the available speed profile.

2.5.1.3 *Criticism of CA Approach*

Again, Hoogendoorn offers some criticism of a proposed aspect of this project. He and Bovy assert that CA models' "oversimplified or incomplete behavioural rules prohibit application to complex situations where the microscopic behaviour of pedestrians is important" (Hoogendoorn & Bovy, 2003). In this project, it is believed that the constrained, corridor-based environment that is characteristic of escalators and moving walkways provides a suitably rigid framework for a CA model to work. The presence of lanes and balustrades means that there is little need to worry about the trajectory and turning behavior of the pedestrians within the system, which is the primary concern of Hoogendoorn and Bovy. Additionally, the natural fit between regular spacing of the escalator treads and cellular framework of the CA model lends credibility to the approach taken in this paper.

2.5.1.4 *Calibration of Model Parameters*

However, there remain several important factors to consider during the construction of the CA model. One paper on implementing car-following procedures discusses issues with calibrating the model and the importance of preventing a behavior they refer to as "particle hopping." This condition exists where platoons of lane changers form in something of a "tailgating dance" that is a manifestation of a "cooperative ping-pong effect" that stems from all users following the exact same decision-making process. Therefore, if there is a series of users following each other closely and the criteria for a lane change are met with all of them, it is possible to see several pedestrians make lane changes back and forth in rapid succession (Rickert, Nagel, Schreckenberg, & Latour, 1996). In order to reduce the occurrence of this sort of event, it becomes necessary to add in complicating factors like delays between lane changes or even some small, randomized probability that a user will decide not to change lanes even when it may be in their best interest to do so.

2.5.2 **Modeling Language**

In order to provide the capabilities needed by the proposed simulation, a program called NetLogo was selected to construct the model. NetLogo is at once a programing language, a compiler, and a simulation software. Based on the storied Logo programming language, NetLogo is designed to provide an object-oriented approach to problem solving. By creating individual simulated entities – called "turtles" – each turtle can be given ownership of characteristic variables that it can use to interact with other turtles and its environment. For instance, a turtle could be designated a "commuter" with some user-defined desired speed that it wants to attain while traveling along a moving belt. Based on these variables and the rules under which the simulation operates, the system as a whole exhibits behaviors approaching those found in the real world. Global variables can then be used to extract system-wide parameters from the model (Wilensky & Tisue, 2004).

In addition to the benefits of the object-oriented framework with individual user "turtles", NetLogo provides another important benefit in that it provides the capability to create a graphical display of the simulation environment. Through this simulation window, the programmer can see the model in action and observe the progression of turtles through the system. However, a more important benefit of this feature is that it allows the programmer to verify that the model is behaving as intended and that the operation does indeed approach real-world behavior. It is through verifications such as this that the calibration of applicable model parameters can be performed, as was described in Section 2.5.1.4, above.

Finally, although NetLogo allows for programming to occur within a CA framework, the software itself is written to allow for actions to occur in a continuous manner. By assigning a continuous, decimal speed to each user rather than a discrete value that is a multiple of the tread spacing, additional points may be created in the speed profile. In this way, users may be able to travel a "half step", putting them astride two consecutive treads as is characteristic of many transit users, as was described in Section 2.5.1.2, above.

Chapter 3 – Model Development

The literature review revealed a plethora of pertinent statistics and principles. Even though little direct research exists on the specific topic of determining the capacity of a moving belt facility through microsimulation, by pulling the necessary information from research conducted in other fields enough data can be collected to construct the proposed model. In this section, the development of the moving belt model will be described including all necessary assumptions and the logical framework contained therein.

In order to make the resulting program easier to visualize and comprehend, the model was split into two primary sections. First, there exists a SETUP routine to construct and initialize the model and its component variables. Depending on the selections made by the user, the model can start either empty or populated with pedestrians based on the inflow that was specified. After being properly initialized, the GO routine is what provides the actual functionality of the model. Finally, several additional protocols have been enacted to facilitate data collection.

3.1 Initialization

In order to properly initialize the model, a SETUP routine was developed. This set of instructions provides all the information that NetLogo needs to adequately define the simulation environment and to configure its initial pedestrian state. The components of the SETUP routine are shown in Figure 3.1, below.

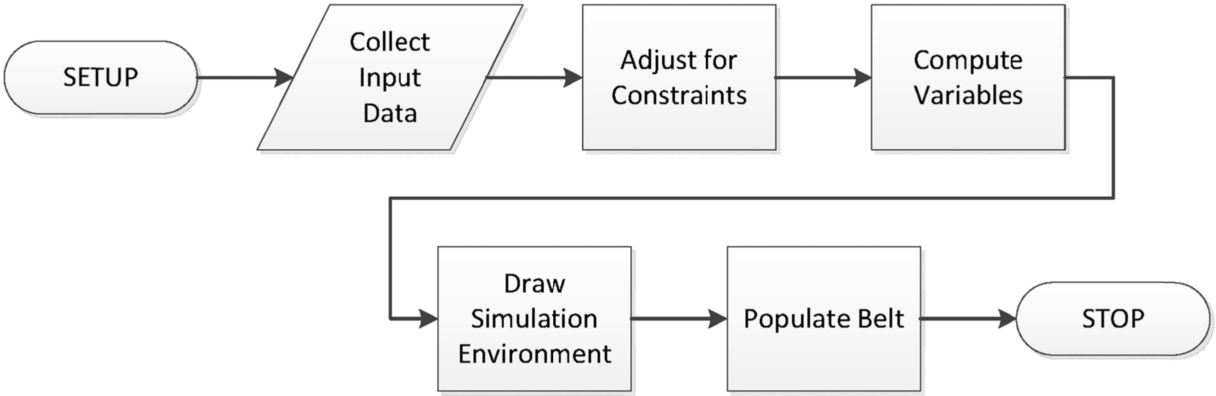
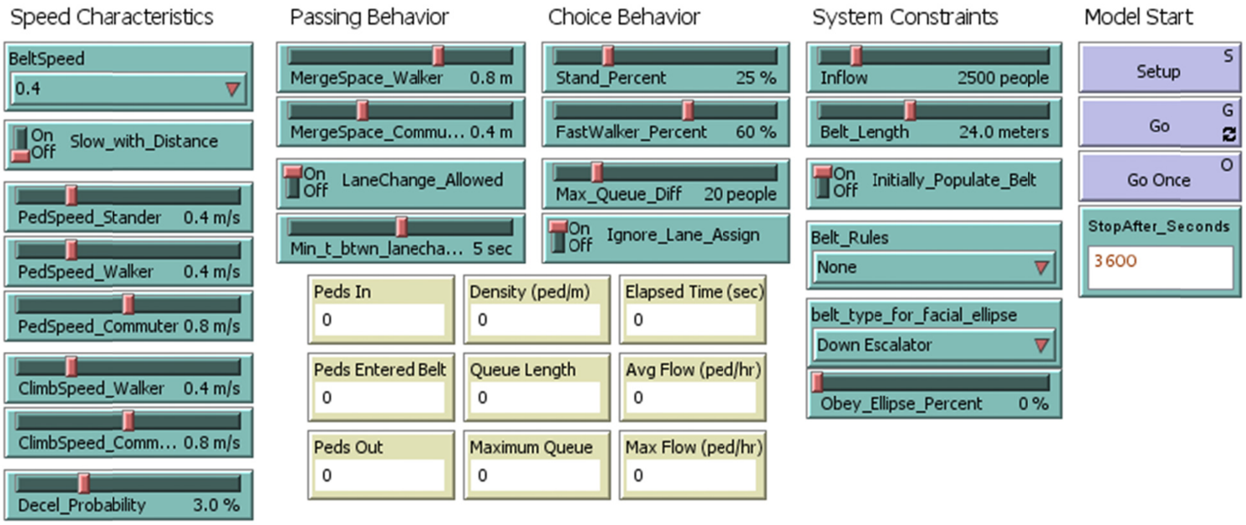


Figure 3.1: Flowchart of "SETUP" Routine

SETUP runs through several sets of processes in the course of initializing the simulation space, but because of how it functions each iteration of SETUP by definition must clear out any settings that existed previously. Therefore, SETUP only needs to be run once before the operation of the model can begin with the GO command.

Each of the modules contained within SETUP has a specific function in the logical progression of model initialization. The first three steps process the user-defined input data and make adjustments as necessary where conflicts may exist, while the following steps define the simulation space. In this section SETUP will be broken down based on the two processes which produce observable results – the ones that define the simulation space – with the processes involved in the previous three modules divided into their role in defining either the simulated belt or in populating the system with pedestrians. In order to provide a better understanding of the inputs available within the model, the control board as rendered in NetLogo is shown as Figure 3.2, below. In this figure, the green boxes represent user inputs, the tan fields are output dials, and the blue buttons represent actions, including the aforementioned SETUP and GO routines.



BeltSpeed 0.4
Slow_with_Distance
PedSpeed_Stander 0.4 m/s
PedSpeed_Walker 0.4 m/s
PedSpeed_Commuter 0.8 m/s
ClimbSpeed_Walker 0.4 m/s
ClimbSpeed_Comm... 0.8 m/s
Decel_Probability 3.0 %
MergeSpace_Walker 0.8 m
MergeSpace_Commu... 0.4 m
LaneChange_Allowed
Min_t_btwn_lanecha... 5 sec
Stand_Percent 25 %
FastWalker_Percent 60 %
Max_Queue_Diff 20 people
Ignore_Lane_Assign
Inflow 2500 people
Belt_Length 24.0 meters
Initially_Populate_Belt
Belt_Rules None
belt_type_for_facial_ellipse Down Escalator
Obey_Ellipse_Percent 0 %
StopAfter_Seconds 3600
Peds In 0
Density (ped/m) 0
Elapsed Time (sec) 0
Peds Entered Belt 0
Queue Length 0
Avg Flow (ped/hr) 0
Peds Out 0
Maximum Queue 0
Max Flow (ped/hr) 0

Figure 3.2: Model Control Board in NetLogo

3.1.1 **Belt Characteristics**

Because of the range of capabilities afforded by the model, the process of defining the simulation environment can require a fair amount of input on the part of the user. Although example scenarios are provided to give a series of demonstrations to outside observers, for someone using the model for research purposes there is a great deal of flexibility in its functionality. In either case, the model requires information about the belt as well as the rules that exist on board it in order to create the initial simulation environment.

3.1.1.1 *Belt Parameters*

The parameters that define a moving belt system generally are those that pertain to its size, overall operation, and type of belt that is being modeled. To this end, the user is asked to input the length of the belt in meters as well as the speed of its operation in meters per second. However, because the model is being run under the principles of cellular automata (CA) operation all of these values must be converted the cell-based units that will be used throughout the model in the "Compute Variables" module. This process requires converting all lengths from meters into cells by dividing by the cell side length, which as previously stated is 0.4 meters along the direction of travel. Since this distance was chosen based on the standard tread depth of an escalator step, these units are called "treads" throughout the model.

The user is additionally able to specify the type of belt to be used in the model. The options available are walkway, up escalator, and down escalator. While escalators are divided based on direction of travel due to the hesitation observed in pedestrians from obscuring of the facial ellipse, walkways are not divided in this manner because their inclination is kept to a low enough level that the facial ellipse is not significantly violated.

An additional capability of the model is the ability to simulate behavior on a staircase – albeit a narrow staircase – by creating an escalator-based simulation with a belt speed of zero. In this situation, code contained in the "Adjust for Constraints" module automatically ensures that there are no pedestrians set to be stationary, as there would be in the case of an escalator or moving belt. It is this type of verification that is contained within the "Adjust" module so that impractical scenarios are automatically flagged or adjusted.

3.1.1.2 *Belt Rules*

Several parameters are also available to the user to specify rules that could be implemented on the belt. One such rule includes the option to specify walk-only or stand-only restrictions onboard the simulated escalator or moving walkway. This rule was provided in order to test the effect of a behavioral prohibition on the overall capacity of the system. The user can also disable lane changing functionality within the model if the experimental scenario calls for the independent operation of the belt's "lanes."

An additional condition that can be altered on board the belt is in regards to the facial ellipse that impacts downhill escalators and staircases. A slider is present on the simulation control board (Figure 3.2) that allows the user to specify how many pedestrians will choose to observe the facial ellipse principle. The remaining users will ignore the desire to leave extra free space between them and the pedestrian immediately in front.

3.1.1.3 *Draw Simulation Environment*

Once all of this information has been collected and processed, the model's framework can be constructed. Commands are sent to NetLogo that define the size of each cell, or "patch," as it calls them, and using the user-defined belt length the program creates a graphic representation of a belt surface as shown in Figure 3.3. In this figure, the blue grid represents the lane- and tread-based cell configuration, complete with shading designed to make the individual cells visible. Along the length of the belt, 40 distinct cells are visible, which at 0.4 meters per tread represents a horizontal belt length of 16 meters. Other purely aesthetic features present in the simulation environment are the black cells, representing handrails, and the light gray cells on either end of the belt showing where arrival queuing and departure dispersion occurs.

Figure 3.3: Simulation Environment as Rendered in NetLogo

3.1.2 **Initial Pedestrian State**

Once the belt has been drawn within the simulation space, the initial pedestrian state can be established. This process involves tabulating and processing the input parameters that relate to pedestrian quantity and behavior. Following this step, the user can choose to have NetLogo automatically populate the belt with a baseline scenario.

3.1.2.1 *Adjust for Constraints*

Using the user-input data, the first thing that must be done is to verify that the data meets with all logical constraints. As discussed before, this step involves performing adjustment of the input data to ensure that any rules included in the model are not broken by user inputs. An example of this situation is seen if the user sets the belt speed to zero, which would imply that a staircase is being simulated. Since a staircase cannot convey its passengers along at speed, the model will automatically upgrade any pedestrians designated as "standers" to the "walker" category, ensuring that the model continues to operate smoothly and efficiently.

3.1.2.2 *Compute Variables*

Once the adjustments have been performed to ensure that all rule sets have been followed, pedestrian variables may be computed from the raw input data. This involves not only converting walking and climbing speed units from meters per second to treads per second, as discussed above; it also includes calculating converting any relevant percentages from whole numbers to their corresponding fractions.

Still further adjustment is required in the calculation of the actual number of pedestrians in each of the three pedestrian classes using the user-defined total inflow rate and the percentage of turtles that is are assigned to each class. Class options are described in Section 3.1.2.3, below. It is important to note that there are only two pedestrian mix sliders present in the model's control panel (Figure 3.2). The first slider controls the percentage of the total inflow that is made up of the walker class while the second shows what percentage of the remaining pedestrians – those who are not standing – will be defined as fast walkers, a class known as commuters. The remaining percentage left on this second slider represents those who are normal walkers as a fraction of those who are not assigned to the stander class.

3.1.2.3 *Pedestrian Parameter Selection*

For this model, three classes of pedestrians have been established to account for different levels of user aggressiveness. At the lowest end of this scale, "standers," as their name implies, choose to stand still on the belt. However, a speed slider can still be seen for these pedestrians in Figure 3.2 since these users still require some speed that they observe while walking along the gray patches that constitute the loading and unloading areas. In the simulation environment, standers are shown as black wedges pointing in the direction of travel.

Next are the "walkers," which possess user-defined levels of climbing speed, walking speed, and required free space behind in the opposite lane in order to make a lane change. Walkers are represented by the color green.

Finally, at the highest level of aggressiveness is the class of pedestrians known as "commuters." This name was not chosen to reflect any level of judgment on transit system commuters except perhaps to show that they have the most reason to be hurried in their travels through ambulatory facilities like transit stations. Consequently, commuters are expected to be set to the most aggressive, speed-based tendencies, meaning higher levels of walking speed and climbing speed along with a lower level of space required to complete a merging operation. Commuters are displayed as red wedges.

3.1.2.4 *Populating the Belt*

Using the previously established pedestrian class parameters, the pedestrian mix present in the simulation environment can be fully defined. On the control board (Figure 3.2) there exists a switch that causes the simulation to automatically populate the belt with pedestrians if desired by the user. When this module is activated, the model creates pedestrians of each class that are randomly distributed along the length of the belt. The quantity of pedestrians that are generated is based on a Poisson distribution centered around the expected value of pedestrians that would be present on the belt under that given belt length and operation speed. The lane in which a given pedestrian is generated is determined based on other user-selected lane assignment options.

Each pedestrian within the system is created with initial speed, merging, and facial ellipse parameters appropriate to its class. Additionally, each pedestrian is set to point to the right since this is defined as the forward direction in the model as well as set to the appropriate lane. At the same time, each pedestrian is initialized with several other count variables to keep track of parameters such as its presently desired speed and the time since it last changed lanes.

An example of the resulting simulation environment is shown below as Figure 3.4. Note that in this model the forward direction will always be to the right, in that pedestrians will enter the simulation from the left and progress forward along the belt until they exit on the right and are removed from the environment and counted. It can also be seen that under the conditions imposed when this figure was generated, all walkers – the black wedge shapes – are at the bottom of the simulation. However, when the direction of travel is considered it becomes apparent that this means they are traveling in the right lane and are pointing forwards along the model. Walkers (green) and commuters (red) can likewise be seen in the left lane.

Figure 3.4: Pedestrian Populated Simulation Environment as Rendered in NetLogo

3.2 Operation

Following the completion of the SETUP routine, the model is ready to run. In order to begin the simulation process, the GO routine must be selected. Two options exist to execute this procedure. The user can either select the "Go Once" button to progress the model through one "tick," the name given by NetLogo to a simulation time step which in this model is one second, or they can select the "GO" button. GO is what NetLogo calls a "forever button," meaning it will continue to run until the user stops the simulation, an error occurs, or the simulation's ending criteria are met.

Figure 3.5 shows a flowchart of all the modules contained within the GO routine. At the end of the chart, a diamond block can be seen. In flowchart notation, ovals represent start and ending commands, rectangles represent actions or modules, and rectangles with bars on the end represent subroutines. However, the diamond block represents a decision structure, usually a true or false decision. In this case, the decision present in the GO routine exists to see if the routine's ending conditions have been met by the end of a given simulation second.

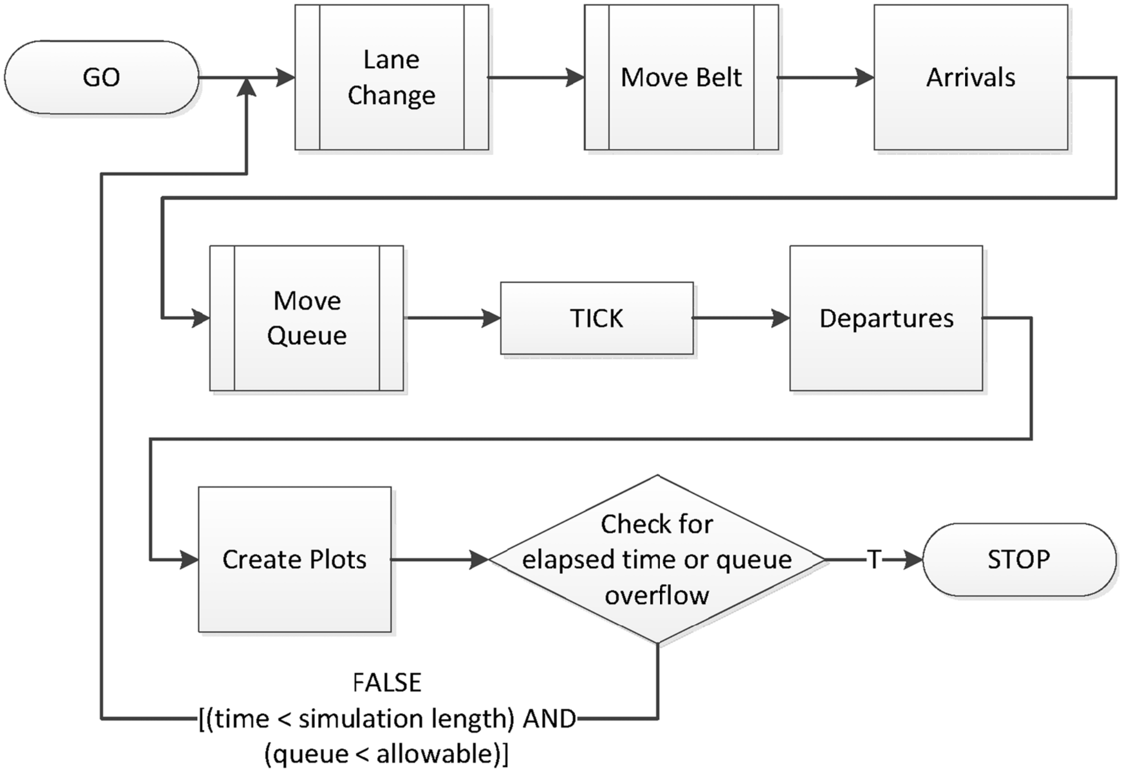


Figure 3.5: Flowchart of "GO" Routine

3.2.1 **Inducing Movement**

In developing the model, the first goal was to induce movement. In early versions of the model, before the implementation of following and passing rules, pedestrians present in the system simply walked unimpeded along their original trajectory at their original speed. This behavior caused pedestrians to pass right through other, slower moving pedestrians because the logical framework that causes a real-world human to slow down upon encountering another pedestrian had not yet been programmed.

However, these early steps of model development were necessary in order to verify that the underling framework of the model functioned satisfactorily before more advanced subroutines were added. While pedestrian characteristics like desired speed had been accounted for the belt population module (Section 3.1.2.4), the fact that the model involves pedestrian movement on a moving belt surface meant that the speed of the belt must be included in addition to the relative walking speed.

The process of computing the total speed simply involves adding the relative speed that a pedestrian achieves through climbing to the belt speed that exists in the simulation. This total speed represents of a pedestrian on the belt as seen by outside observer, whereas the relative speed is just that – relative to the movement of the belt – so this speed is as if it were observed by someone else standing stationary on the belt.

With all the passengers on the belt traveling at their appropriate total speed, there remains one more factor that must be accounted for before additional process modules can be implemented on the system. Although it may seem as though all pedestrians created in Section 3.1.2.4 contain all the necessary parameters and behavioral characteristics for model operation, in reality all that has been programmed into them thus far is their desired speed. An additional variable needs to be included in the framework of every pedestrian that represents their current speed, since any obstruction – like a slower walker on the belt – will by rule cause a conflict between the user-defined speed of the pedestrian and the amount of distance that the pedestrian is capable of advancing.

In this way, even if an entity on the belt is slowed down for some reason, they will still have enough "knowledge" to know what speed they should accelerate back up to once the obstruction is passed. However, since at this point in the model development following behavior (Section 3.2.3.1) has not been implemented, the current speed variable will be initially set to the desired speed value.

With the above logic in place, model development could finally progress to additional modules. The speed logic contained within this section has been included in the detailed movement procedures contained in Section 3.2.3.1.

3.2.2 **Boundary Operations**

Once movement of the simulated pedestrians was achieved, it became time to implement a number of boundary operations to deal with the interface between the belt and the simple walking environment. As shown in Figure 3.5, the modules that represent these conditions involve pedestrians entering the system and arriving at the belt, pedestrians departing the belt and being removed from the simulation environment, and finally the procedures by which the GO routine itself is terminated.

3.2.2.1 *Arrivals*

In order to assure continued operation of the model in successive time steps, new pedestrians must be introduced to the simulation space. This function is performed by the arrivals module. In this module, pedestrians are generated at the entry end of the belt and assigned parameters in a process very similar to the belt population procedure described in Section 3.1.2.4, above. Each entity is assigned a heading and lane based on its pedestrian class along with a desired speed based on its walking capabilities and an initial speed equal to this desired speed. Depending on the queue situation in that simulation second and whether the user has instructed the model to ignore lane assignment under certain unbalanced conditions, the pedestrian may instead be assigned to a random lane.

As before, the quantity of pedestrians within each class is determined by the total number of pedestrians within that class that will be generated within a one-hour simulation, as was calculated in Section 3.1.2.2. Specifically, in each second the number of generated pedestrians within a given class is defined by a Poisson-distributed random number centered around a value equal to the number of one-hour arrivals divided by 3600 seconds. In practice, the sum of these one-second Poisson-random arrivals value nearly always sums up to the original one-hour arrival quantity, as would be expected of a sum of Poisson-distributed random numbers.

Once generated, the pedestrians are adjusted backwards to ensure that they are in the back of the queue, should one exist. These pedestrians will later be governed by the "Move Queue" procedures discussed in Section 3.2.3.1.

3.2.2.2 *Departures*

Just as pedestrians must be inserted into the simulation environment, they must be removed from the simulation environment to approximate real-world behavior. The departures module checks the belt system for any pedestrians that have progressed past the end of the belt treads and removes them from the model space using NetLogo's DIE command. As they are removed, several global count variables corresponding to the various pedestrian classes tabulate the exiting pedestrians. If this module is inspected (complete model code is contained in Appendix A), it can be seen that this module also contains some computation of system-wide variables for use in the model's data collection operations.

3.2.2.3 *Ending Conditions*

The simulation can be terminated in one of several ways. Many of these methods were observed during the model development, as any error message dealing with the execution of the simulation's calculations will halt the GO routine. For instance, if the model comes upon a section of unexecutable code or has to perform an undefined mathematical operation like dividing by zero, then the simulation will stop. However, the more common procedure by which the model exits its iterative execution procedure is contained within the diamond block present in the GO routine flowchart (Figure 3.5). This module represents the logical decision between continuing the model for another iteration of one simulation second or stopping the GO routine after that particular cycle. The decision made at this point in the routine is based on the exit conditions that govern the model.

There are two exit conditions for the GO routine in this model. The first condition checks to see if the user-defined maximum queue has been reached, which would mean that the queue has grown to be so long that it is approaching the edges of the simulation environment. This condition is known as queue overflow, and if the queue were to reach the edge of the simulation window it could cause NetLogo to crash. The second condition checks that the elapsed time of the simulation, as measured by the cumulative number of ticks that have been tallied by the model, does not exceed the user-specified simulation length. As can be seen in Figure 3.5, each iteration of the GO routine increases the tick counter by one through the inclusion of a TICK module. Therefore, it can be said that GO will perform the same set of procedures every simulated second until the exit conditions are reached.

3.2.3 **Implementing Pedestrian Interaction**

Although great progress had been made up to this point in the development of the model, the configuration still did nothing to prevent pedestrians from simply walking through each other. Clearly, rules of pedestrian interaction were needed before the model could be checked against empirical capacity values. To perform this function, the following and passing rules for highway traffic systems found during the literature review (Section 2.2) would need to be implemented within the context of this pedestrian model.

3.2.3.1 *Following Behavior*

Following rules were implemented first. As can be seen in the GO routine flowchart (Figure 3.5), there are two modules that require pedestrians to move, specifically "Move Belt" and "Move Queue." As their names imply, Move Queue involves the progression of pedestrians waiting to board the belt surface while Move Belt governs the movement of pedestrians who are have already boarded the belt. The general procedures involved in the two following behavior modules can be seen in Figure 3.6.

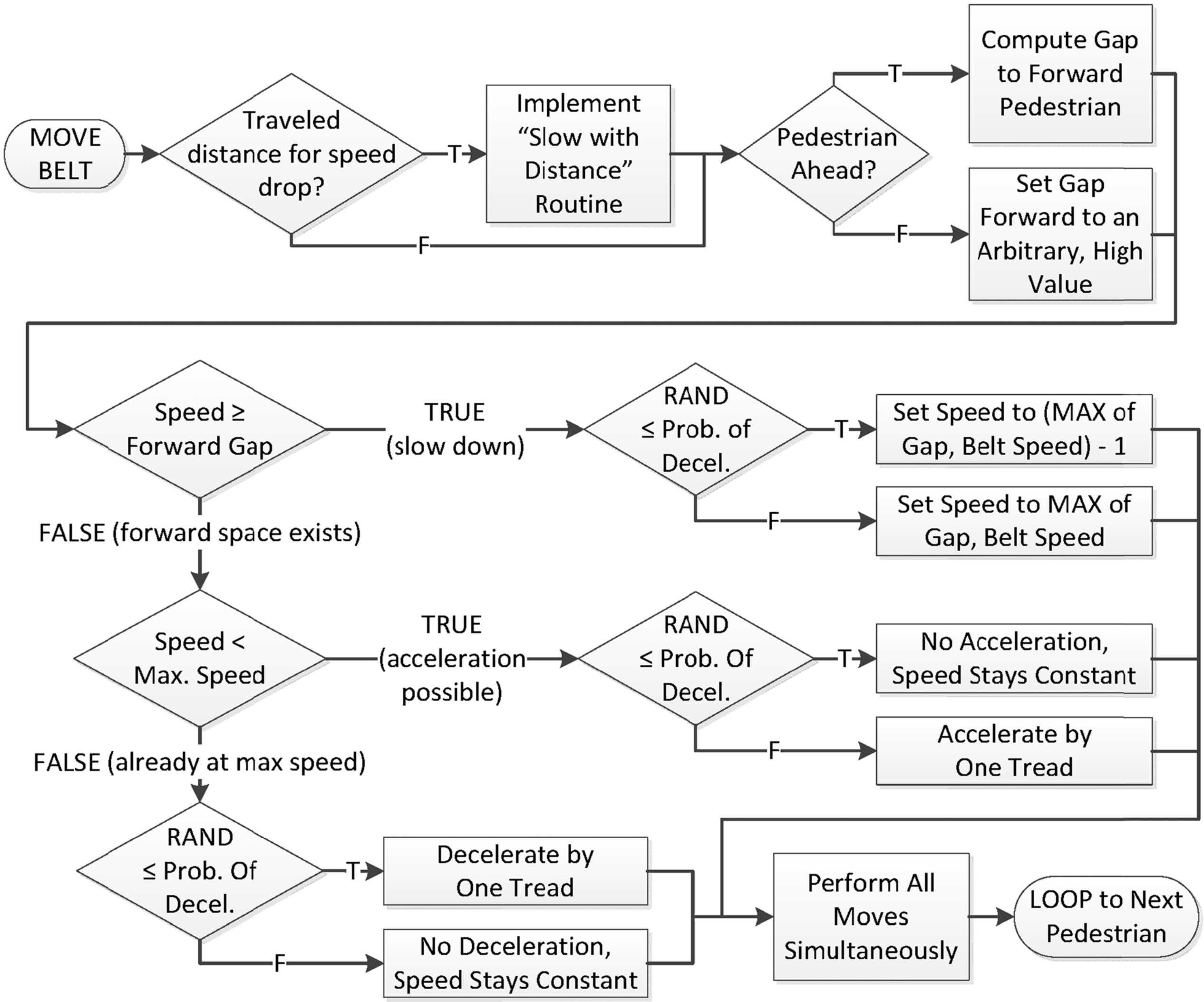


Figure 3.6: Flowchart of "Move Belt" Subroutine

It is important to note that the procedures shown in Figure 3.6 relate specifically to the Move Belt subroutine and that slight differences exist between Move Belt and Move Queue. Specifically, since Move Queue applies in the horizontal waiting space before entering the belt, pedestrians are not subject to fatigue as they would be when climbing a stairs so the initial "distance traveled for speed drop" decision and resulting "Slow with Distance" subroutine implementation (Section 3.2.4.1) is ignored.

Although all other logic remains in place, since Move Queue occurs by definition within the context of a queuing situation in most cases the subsequent decisions will consistently yield little to no headway and consequently no ability to accelerate. Therefore, forward progression will by and large stem from gaps that open as pedestrians are able to board the belt at the interface between the queue and the belt.

An additional factor that must be considered at the interface between the queue and the belt is the change that pedestrians experience between their desired walking speed and their desired climbing speed. Once a pedestrian crosses onto the belt, the model marks that pedestrian as "on belt" which triggers several actions. First, the pedestrian's desired speed is changed from their maximum walking speed to their maximum climbing speed. In the case of the stander class, this means that their desired speed is instantaneously set to zero. For other classes, this means that their maximum desired speed will decrease; however, since this is speed is relative to the movement of the belt their total speed will likely increase. This increase in average speed is what keeps a queue from developing at the base of the belt in unsaturated conditions. Another function of the "on belt" marker is that it tells the model that that particular pedestrian is now subject to the Move Belt and Lane Change subroutines rather than Move Queue procedures.

3.2.3.2 *Passing Behavior*

After the implementation of following rules, the operation of the model was observed to degrade significantly from the previous iteration of the model. However, this drop in performance of the simulated belt system was necessary to approximate real-world behavior, and it in fact served to bring the observed throughput of the model's simulated escalators and moving belts closer to empirically determined capacities from the various studies found in Section 2.4.1 of the literature review.

However, one final step still remained, and that was the implementation of lane changing rules. Although it is common practice on escalators and moving walkways that those remaining stationary relative to the belt should stay in the outside lane with those pedestrians actively walking or climbing along the belt sticking to the inside, in reality people do not always adhere to these guidelines. Therefore, the "Lane Change" subroutine was developed in order to provide a structure to pedestrian passing within the simulation. The logical steps contained within the Lane Change module are shown in Figure 3.7.

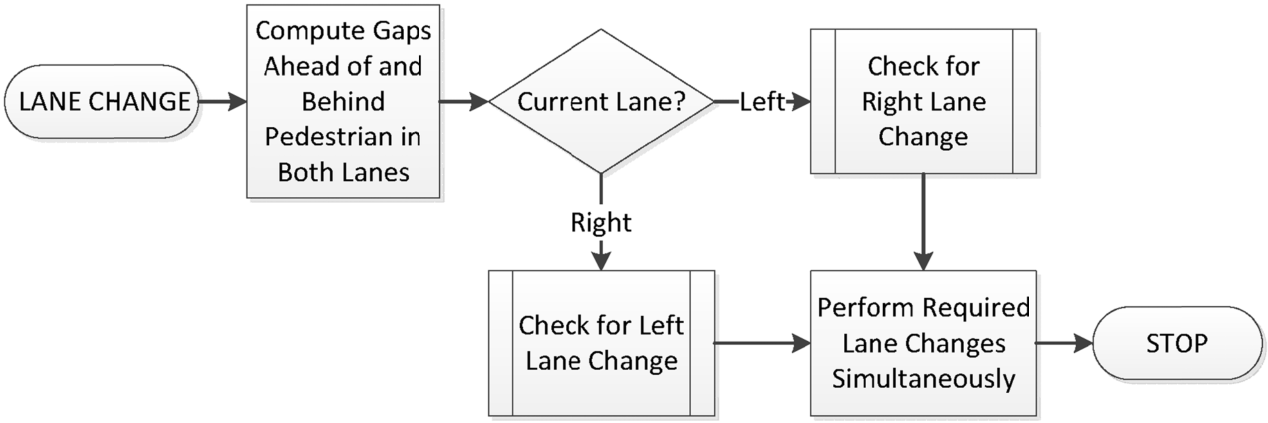
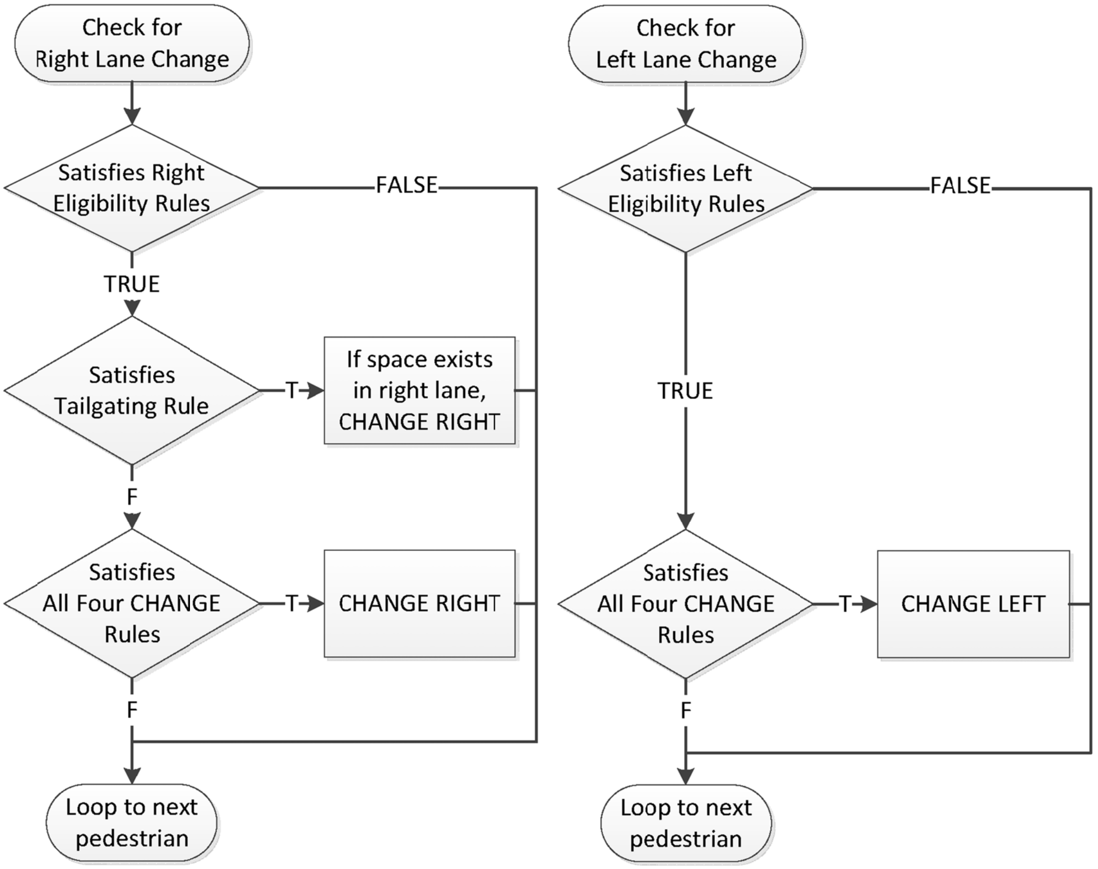


Figure 3.7: Flowchart of "Lane Change" Subroutine

The Lane Change subroutine begins by calculating the open space around each pedestrian in the system, both ahead of and behind the pedestrian in both the current and adjacent lane. Next, one of two procedures is run to determine if a lane change is warranted, depending on which lane the pedestrian is currently traveling in. These procedures are shown in Figure 3.8. Also included in this figure are the three sets of rules that are used by the lane change check algorithms to determine which pedestrians will benefit from a lane change. First, the pedestrians are scanned to ensure that they are in the appropriate lane and that they have not recently changed into that lane. Second, pedestrians eligible for a right lane change are checked to see if they are being "tailgated" by a faster pedestrian who is following closely behind. If this is the case, they are immediately flagged for a lane change. Finally, parameters for the remaining eligible pedestrians are passed through four tests based off of the TRANSIMS lane change rules. If all four conditions in this set are met, the pedestrian is flagged for a lane change.



Right Eligibility Rules:
Pedestrians must be in left lane, on belt, and passing minimum time between lane changes

Tailgating Rule:
Faster pedestrian following closely behind

Left Eligibility Rules:
Pedestrians must be in right lane, on belt, and passing min. time between lane changes

CHANGE Rules:

- Speed Desired > Space Ahead
- Gap Forward in Opposite Lane > Gap Forward in Same Lane
- Gap Back in Opposite Lane > Tolerable Gap Back in Opposite Lane
- Gap Forward in Opposite Lane > Speed

Figure 3.8: Flowchart of Lane Check Algorithms

After the model has evaluated all pedestrians and flagged the ones who would benefit from a lane change, all pedestrians who qualified for a shift are moved to the adjacent lane simultaneously. The reason that the lane change check algorithms are performed on all pedestrians on the belt before any pedestrians are moved is so that no pedestrians gain an advantage in their decision process as a result of their position on the belt or any other arbitrary parameter. Therefore, all pedestrians make their decisions based on the gaps around them at the same instant in time, thereby ensuring that an open gap for a walker will not be filled by a faster moving commuter before the walker is allowed to make a decision, for instance. In this way, the GO routine sequence and protocols (Figure 3.5) are maintained.

3.2.4 Accounting for Complicating Factors

Even after implementing pedestrian interaction protocols in the model, there still remained a number of other complicating factors that had previously been identified in the literature review (Section 2.1.2) that needed to be addressed.

3.2.4.1 *Pedestrian Limitations*

The first complicating factor to be implemented in the model came from the pedestrians themselves. Humans naturally will tire under the strain of prolonged physical exertion, so a module was added to the GO routine to cause the simulated pedestrians to exhibit the symptoms of tiredness. This behavior is manifested in the model in a reduction of the desired speed for a given pedestrian once they have traveled a certain distance. In this model, the "Slow with Distance" routine (shown in Figure 3.5) performs this function. When enabled, the routine causes pedestrians to slow by one tread per second after traveling 20 meters horizontally along the belt and another tread per second for every additional 10 meters, as suggested by the literature.

However, in order to implement this module a distinction needed to be made between a pedestrian's initial desired speed and the speed they desire at a given moment in time. In this way, the "speed desired now" variable was initially set equal to the "speed desired" by that pedestrian, but after 20 meters the equality was altered so that the speed desired now was instead equal to the speed desired minus one tread per second. After 30 meters, this became speed desired minus two treads per second, and so on. This process was implemented to make the subroutine more intuitive and easier to program.

3.2.4.2 *Bottleneck Factors*

Another critical factor shown in the literature, especially in the Hoogendoorn experiments, is the importance of modeling the impediment to travel that exists at a bottleneck. In many of his corridor experiments, Hoogendoorn observed a noticeable delay in pedestrian travel when a substantial volume of pedestrians was funneled down through a bottleneck (Hoogendoorn & Daamen, 2005). Therefore, it is imperative that this model should exhibit a comparable level of hesitation from both the bottleneck and the interface that exists between stationary floor and moving belt.

However, when comparing the operation of the model and the resulting capacities determined therein, it appears that this simulation does indeed approximate the queuing and delay incurred by travelers as they traverse the system based on the capacities shown in the literature (Section 2.4.1). It is believed that this similarity has two causes. First, although Hoogendoorn's experiment featured pedestrians converging on the bottleneck from numerous directions and only merging at the entry to the bottleneck, pedestrians and especially transit commuters exhibit a common set of courtesies to one another in their normal routine, and one of these is queue discipline. Transit commuters have been observed to form distinct queues at the base of escalators under crush loading conditions while attempting to exit transit stations (Davis & Dutta, 2002), much like the behavior exhibited by the pedestrians in this model.

Furthermore, the delay caused by pedestrian hesitation at the interface between a stationary floor and the moving belt is approximated quite well by a completely inadvertent feature of the model. Since the transition of each pedestrian between the Move Queue and the Move Belt procedures causes them to verify that a suitable gap exists on the belt ahead of them before boarding, some pedestrians are delayed ever so slightly in that they may have to wait a second before finding a gap that fits their needs. This occasional delay very closely approximates the behavior that is observed when a diverse stream of pedestrians attempts to board a moving belt facility.

3.2.4.3 *Belt Parameters*

Still another complicating factor that has been implemented in the model deals with the belt parameters themselves, as was implemented all the way back in the model initialization phase of development. It is important to recall that the literature review revealed that facial obscuring (Section 2.1.2.2) can cause discomfort to stair climbers or pedestrians traveling upwards on an escalator if they follow too closely behind the passenger in front of them. In order to account for this tendency in the model, some amount of space must be reserved in front of pedestrians who choose to observe the facial ellipse concept.

Since the user is able to define whether a belt is traveling in the upwards direction as well as the percentage of pedestrians who choose to observe the facial ellipse (Figure 3.2), these preferences can be implemented in the model. First, extra code was added in the Populate Belt and Arrivals modules to determine which pedestrians – if any – would be randomly selected to observe the facial ellipse through a combination of random probabilities and the belt type present in the simulation. If these conditions were satisfied, the pedestrian would be assigned a facial ellipse size of one tread. If not, the facial ellipse variable would be set to zero.

The Move Belt routine was then modified to include an additional distance equal to the value of the facial ellipse variable to each computed headway term present in the logical decision tree. In this way, a pedestrian who observes the facial ellipse concept will begin to slow earlier in order to preserve one tread of spacing between themselves and the forward pedestrian when traveling on an uphill facility.

3.3 Data Collection

With the model performing in a way that accurately portrays realistic behavior, steps must be taken to ensure that analyses can be performed on simulations that are conducted using the model. This means that data can be collected following a simulation run as well as at regular intervals during the run. To do this, several outputs were constructed along with several subroutines that permit the user to expand the functionality of the model as a whole.

3.3.1 **Real-Time Outputs**

As was seen in the NetLogo-rendered control board in Figure 3.2, several real-time outputs were placed on the control panel for easy viewing. In addition to displays showing tallies of the number of pedestrians who had entered and exited the system through that instant in time, the results of other computations contained within the "Create Plots" module of the GO routine (Figure 3.5) were shown. These calculated parameters included the current and maximum observed queue lengths, as well as the density and instantaneous flow rate exhibited by the model. A running average flow rate was also displayed once the model had been given time to stabilize into a steady condition, which was defined as occurring after the belt had a chance to clear itself of stationary pedestrians five times. The last real-time display "dial" showed the number of simulated seconds that had passed since the start of the simulation.

Beside the array of dials, two plots were also included to allow the user to track the progression of these and other variables throughout the simulation period. In addition to the previously mentioned density, flow, queue, and pedestrian count variables described above, a second plot was created to show the average speed of each class of pedestrians for all members presently in the system. This plot was included to show the level of impedance that the walker and commuter classes experienced through a comparison between the user-defined defined speed and the observed average speed.

3.3.2 **Time-Space Diagrams**

Another method to observe the progression of users through the system and to determine the level of impedance experienced as a result of obstructions like the bottleneck, the floor-belt interface, and slower pedestrians is through a time-space diagram. This type of plot shows the trajectory of a single entity within the system, with the elapsed time on the x-axis and the distance traveled displayed along the y-axis. Because of this setup, the speed of the traveler can be determined at a glance by observing the slope of the produced line at any point in time.

Procedures were implemented in the model that permit the user to track up to three pedestrians at any one time in a real-time time-space diagram. The model is capable of randomly selecting three pedestrians; however, the user also has the option of overriding these random selections by enabling the "Select Ped" subroutine and choosing up to three of their own to track in a third plot.

Unfortunately, the data contained within the NetLogo produced plots is trapped within the model, making this information impossible to export in this format. Additionally, the plots themselves are difficult to interpret, especially over a long time interval. To overcome this issue, a "File Output" subroutine was created that gives the user the option to export all relevant model data to an external text file for analysis or plotting in another program. If activated, this external file will report the simulation system inputs for the purposes of repeatability. Additionally, at one-second intervals NetLogo will print the results of the various performance variables as well as the position, speed, and desired speed of every pedestrian within the system. Using this information, all the plots produced in NetLogo can be reproduced outside of the program.

3.3.3 **Capacity Analysis**

Finally, since one of the major purposes of this model was to determine the capacity of a moving belt system under different input stream characteristics, a final subroutine was added to allow the user to conduct a "Capacity Test" of the entire system. When this module is activated, the simulation automatically takes precautions to ensure that queue overflow cannot occur. To do this, the model monitors the queue length to see if it approaches the maximum queue. If it passes the warning value – by default, 75 pedestrians in either queue lane – the system automatically removes the extra members of the queue. However, before it does this it tallies the offending queue members into a global "queue removed" count variable. This quantity of pedestrians removed from the queue is added to the queue length in all displays and plots featuring queue length.

The reason for ensuring that the model will never experience queue overflow when performing a capacity test is so that the inflow rate can initially be set to an arbitrary high value on the first test run under a new set of parameters. Even with the inflow set to a high value, the belt will still only be able to pass a certain number of pedestrians no matter how many are waiting in line. Therefore, although the inflow is well in excess of the observed flow exiting the queue, this observed flow could then be used as the inflow in the second iteration of the simulation. The outflow is continually used as the inflow in subsequent iterations of the capacity test until the system reaches a point where the queue is observed to reach steady-state conditions. At this point, the inflow and the outflow will be equal and the system will be at its maximum sustainable capacity.

The capacity level determined using this procedure should be the maximum flow where the input and output volumes match, meaning that the queue should act in a relatively stable manner. An example of this queue behavior can be seen in Figure 3.9, below. This plot is the result of a simulation performed on an uphill belt with 50% of pedestrians observing the facial ellipse at an inflow of 4300 pedestrians per hour, which was the value computed by the Capacity Test module to be the maximum sustainable flow under these conditions. Although the queue length can be seen to rise and fall with the stochastic nature of arrivals under the model, it remains relatively stable at around 20 people in both lanes, with times of longer queues and periods with no queue at all. The important behavior to note in the plot, however, is that the queue length does not grow constantly over the course of the entire hour modeled in this simulation, which would have been the case in an oversaturated simulation environment.

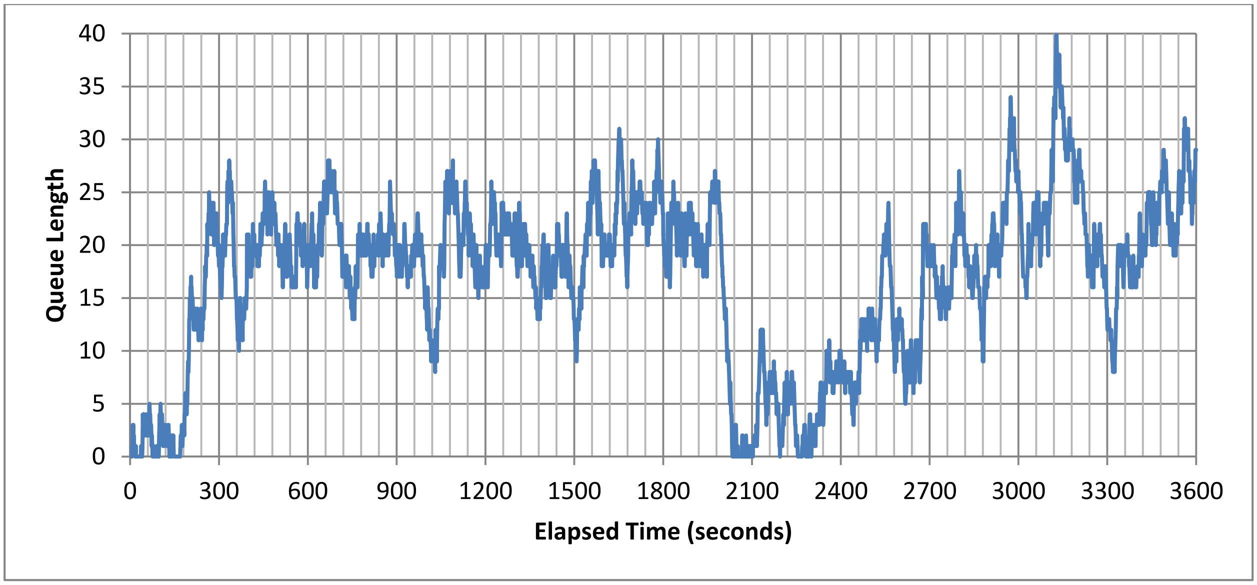


Figure 3.9: Plot Showing Queue Growth under Steady-State Conditions at Capacity

Chapter 4 – Results and Discussion

With the model fully developed and in working order, it could finally be utilized to simulate pedestrian behavior in moving belt systems. As will be seen below, the model has the capability to answer various research questions proposed at the beginning of the project within a controlled, repeatable framework. Following a series of observations designed to validate the performance of the model, both in terms of its representation of pedestrian choice behaviors as well as in its ability to match empirically determined capacity values under the same input parameters, several additional applications were tested.

While the purpose of the model, as it was originally conceived was to compute the capacity of a moving belt system under regionally determined input parameters, the model will be seen below to be capable of conducting several other analyses. The applications investigated in this section include investigating the sensitivity of a proposed belt solution to changes in its inputs, analyzing belt systems under proposed operational rule sets, and determining the required platform size to handle queuing under crush loading conditions.

4.1 Model Validation

In order to verify that the model created during this project accurately represents human choice behavior and the real-world conditions observed in a moving belt system, a series of tests had to be conducted. These validation procedures were implemented to check that the diverse set of behavioral source data as well as the procedures and rule sets pulled from a wide range of transportation research indeed provided a suitable basis for the simulation.

There were two steps involved in this validation process. First, the operation of the model was assessed through a number of visual means to ensure that the simulated pedestrians observed practical behaviors in terms of following behavior, passing choice, and in the level of delay present at the floor-belt interface as a result of bottleneck and boarding effects. After the operational practices had been verified, the model's output performance characteristics could be compared to the capacity that was observed under empirical studies found in the literature when run under the same input parameters.

4.1.1 **Operation**

Based on information and data collected during field studies that were found in the literature, the observed operational behavior of the model could adequately be compared to empirically determined human behavior with the goal of verifying the model's operation. The data that was gleaned from the literature was not limited to numerical results; rather, many examples of diagrammatic and photographic examples of pedestrian behavior were contained within the sources found during the literature review.

In this way, queuing regimes and pedestrian flow formations could be observationally verified. The visual manner by which this validation process was conducted occurred in much the same way as the following and passing behaviors, which were verified through an inspection of the processed by which lane changing was executed and headways were maintained in the face of obstructing pedestrians.

4.1.1.1 *Following Behavior*

Although no video evidence was found in the literature, following behavior is common enough in everyday life for this verification step to be performed based on personal experience. The main validation step that needed to be conducted was to ensure that the programmed following logic matched up with that which was found in the TRANSIMS literature and then subsequently programmed into the model and then to verify that normal following behavior occurred.

First, the model was observed in operation under a wide range of input parameters and operational rule schemes to ensure that the following rules were executed properly. Fortunately, the object-oriented nature of NetLogo means that these rules were graphically displayed in the simulation environment, much as they are seen in Figure 4.1, below. This figure provides the opportunity to validate that these TRANSIMS rule sets are observed by the model as well as to visually verify that they match real-world conditions.

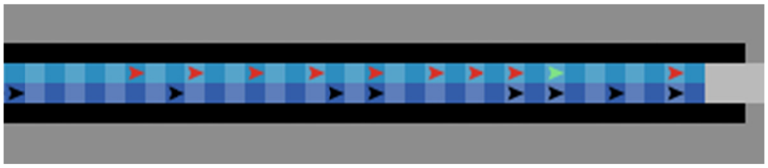


Figure 4.1: NetLogo Simulation Showing Following Behavior

For instance, in the simulation second shown in Figure 4.1 a platoon of commuters, in red, can be seen forming in the left lane – at the top of this figure – behind a walker, in green, because of the lower climbing speed assigned to the walker. Since it is known that the direction of travel in the simulation is from left to right across the screen, commuters just entering this platoon are at the leftmost edge of the figure, with the leading walker closer to the right of the figure. It can be seen that those commuters just entering the platoon are observed to leave a larger gap – two treads – between themselves and the commuter in front than those commuters further along the belt, who only leave one tread.

The reason for this behavior is because the rearwards commuters are still traveling at a higher speed, and therefore the TRANSIMS logic is causing them to leave a larger gap forward. The commuters farther ahead along the belt have had to slow down because of the walker, and because of this drop in speed only need to leave one empty tread to account for any sudden drops in speed they may encounter. This behavior is similar to how automobile drivers reserve larger forward headways at higher speeds to account for the necessary reaction and deceleration time required in that situation.

It is also interesting to note that in the simulation shown in Figure 4.1 the belt direction was set to downhill, meaning that the facial ellipse was not a factor. These gaps observed in this figure exist solely to leave deceleration room in the case of a sudden stop on the part of the leading pedestrian. The facial ellipse concept becomes important when users are tightly packed onto a belt in saturated conditions, as those pedestrians who choose to observe the facial ellipse effect will be compelled to leave an empty tread in front of themselves for personal comfort.

In the course of conducting this evaluation process, an additional effect was observed when the Slow with Distance module was activated by the user. Because of the means by which users slow down in the real world as a result of fatigue, the effective throughput of which a belt is capable decreases as its length increases. This is because a longer belt will see its riders tire as they progress forward along its length, causing a corresponding drop in average relative pedestrian speed towards the unloading end. Above a certain flow rate, this speed drop will cause a shockwave to form and travel backwards down the length of the belt. Beyond this shockwave front are users who are constrained by the decelerated pedestrians. Therefore, even though walkers and commuters do not slow until they near the end of the belt, other pedestrians traveling further back will still experience delays as a platoon of slow moving riders forms behind these fatigued pedestrians.

4.1.1.2 *Passing Choice*

Because of the level of personal preference present in making lane change decisions, more rigid verification of these procedures was required. However, because of the lack of literature regarding passing behavior on stairways and escalators as well as within constrained passageways like moving belt systems in general, the primary focus of this validation was in ensuring that the TRANSIMS passing rules were followed. To this end, a similar process was undertaken as was seen in the validation of following behavior in Section 4.1.1.1, above. Specifically, the behavior of the model under a wide variety of operational conditions was visually inspected for conformity. An example of the model that highlights passing behavior is shown in Figure 4.2.

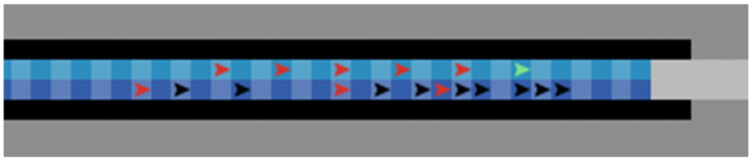


Figure 4.2: NetLogo Simulation Showing Passing Behavior and Gap Acceptance

In this figure, the left travel lane at the top of the figure is limited by a walker (green) who is completing a pass of a platoon of standers (black wedges) in the right lane. However, because walkers are defined with a slower climbing speed than commuters (red) are, a queue of commuters has formed behind the leading walker. Their desired speed indicates that the commuters want to go faster but cannot make a pass to the right because of the platoon of standers in that lane. In this scenario, passing decisions can be understood through the presence of the three commuters present in the right lane, who will now be inspected one at a time.

The commuter located farthest back on the belt – that is, closest to the left of the figure – was observed to make a change into the left lane in next simulation second. This behavior is logical because this pedestrian meets the minimum eligibility rules of being on the belt and having free space open to the left in addition to all four of the TRANSIMS lane change criteria outlined in Figure 3.8. The first of these criteria, a lack of free space in the current lane, can be seen in that there is only one open cell in front of this pedestrian. Secondly, there can be seen to be more free space available in the opposite lane. Next, traffic in the adjacent lane is observed to be moving fast enough to permit an improvement in speed in the left lane. Finally, adequate free space is available behind in the left lane to allow a merge while satisfying the preferred free space required by a commuter of three free treads under this operational scenario. Since all of these tests are passed, a lane change will be conducted at the appropriate point in the overall structure of the GO routine.

Next, the middle commuter in the right (bottom) lane of Figure 4.2 is inspected. This pedestrian was observed to have recently changed lanes to the right in an attempt to pass the commuter in front of him in the left lane since that commuter's speed was limited by the leading walker. However, as happens in the real world, he did not look far enough ahead before merging right and was therefore blocked by the platoon of standers before he could pass the source of his slowdown.

Finally, the forwardmost commuter in the right lane of Figure 4.2 has been stuck at that position in the stander platoon since he entered the belt nearly twenty simulation seconds ago. In that timeframe, no acceptable gap has yet presented itself in the adjacent lane to merge left, and so he has been trapped in the slow lane for the entire journey. This behavior indicates an additional factor that merits further research (Section 4.3), that of increasing aggressiveness exhibited by users as the amount of delay they have incurred increases. Drivers and pedestrians in the real world have been observed to change their behavior in the face of prolonged obstruction, so future study could examine this effect on moving belt systems to see if passenger preference changes with time.

4.1.1.3 *Floor-Belt Interface Effects*

The final area of model operation that must be inspected for real-world functionality is the point of interface between the stationary floor and the moving belt surface. This area is shown in Figure 4.3, below. As has been discussed previously, there are two significant impacts that this location in the model has on pedestrian behavior.

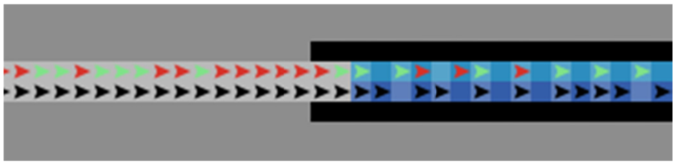


Figure 4.3: NetLogo Simulation Showing the Floor-Belt Interface

First, the hesitation observed in pedestrians when it comes time for them to transfer between the floor and the belt is directly observed in this model in how pedestrians use their own internal decision process to judge when an accepting a gap present in front of them to board the belt. Although this behavior is based on a combination of pedestrian observance of the facial ellipse and the TRANSIMS following rule set, in practice it seems to closely model the behaviors and decisions made by actual pedestrians using real belt systems.

The second cause of delay at the floor-belt interface results from the fact that the entry to a belt is by definition a bottleneck in that it represents a constriction from free movement on a transit platform to the narrow belt. The belt itself provides rigid boundaries to the movement of its riders through the vertical balustrades that hold the system's handrails in position. In typical belt systems, these balustrades are usually separated by a distance of one meter.

In experiments by Hoogendoorn and validated by others, bottlenecks like the ones at the entry to a belt were observed to have a significant effect in limiting the rate at which pedestrians can pass through a constricted system, as seen in Section 2.1.2.3. However, as was discussed in Section 3.2.4.2, it is believed that the set of informal rules that governs transit passenger queuing behavior provides a framework that is similar to that used in the queuing regime of this model.

Since transit passengers typically queue in a relatively linear fashion at the entrance to an escalator or moving walkway (Davis & Dutta, 2002), the effects of the bottleneck are minimized since the pedestrians only have to constrict from the narrow queue into the belt. This is as opposed to constricting into a corridor from a wide area of free space as was measured in the Dutch experiment (Hoogendoorn & Daamen, 2005).

One area of future research (Section 4.3) proposed after the course of this project involves developing a module for implementation in the model that incorporates pedestrian wayfinding in crowd scenarios. This module could then be used to model pedestrian behavior in merging onto the belt by way of the bottleneck created by the balustrades.

4.1.2 **System Outputs**

Following the verification of the model's operation in the previous section, it became time to assess the validity of the various outputs produced by the simulation. Even with the operation of the model having been successfully checked against both real-world behavior and experimental observations found in the literature, if the outputs produced by the model do not similarly match their experimental counterparts then the model itself could not be labeled as a success. To this end, this section will investigate the accuracy of the model's computed capacity values relative to those determined in empirical studies. Attention will also be paid to the model's output data routine and especially to how the resulting data can be used to investigate pedestrian trajectories.

4.1.2.1 *Comparison to Empirical Capacities*

Perhaps the most important validation that needed to be performed before the model could be approved was that capacity. As was mentioned before, the model was created to simulate the behavior of pedestrians on a moving belt system, with one of the major end goals being the determination of that belt's capacity under a user-defined set of input parameters. However, for this goal to be satisfactorily met, the belt capacities determined by the model through simulation had to be checked against empirical sources found in the literature.

Even though one of the stated goals of the model was to separate the practice of moving belt design from the existing methods of utilizing capacity curves and empirical data sources, the accuracy of the simulation compared to real-world installations must be ensured. It is only after the proposed model has been properly verified that a mathematical basis can be brought to the process by which moving belts are selected and specified for use in ambulatory facilities.

In order to use the model to calculate capacity through simulation, the Capacity Test subroutine (Section 3.3.3) had to be activated. With the capacity test switch set to "on," the model could be initialized with the input parameters set forth in the literature. While the studies found in the literature review were clear about the parameters under which the belt was operated, it was found that several studies provided insufficient data about important characteristics like pedestrian speed, aggressiveness, and choice behavior to completely set the inputs of the model as described in Chapter 3. In these cases, average values of normal human exertion and assumptions about choice behavior were used, sometimes in addition to values from other studies conducted in the same geographical region. For instance, a study on climbing speed conducted at a London-area university (Fujiyama & Tyler, 2004), described in Section 2.1.2.1, was used to provide additional information to a capacity study conducted on escalators in the same system (Davis & Dutta, 2002) featured in Section 2.4.1.

4.1.2.1.1 Capacity Scenario 1

The first scenario that was investigated was based on Fruin's measurements on a downhill belt. By setting the model to 20% standers and 60% of the walkers being commuters with lane change allowed and no rule restrictions, a capacity of approximately 6600 pedestrians per hour was observed on an escalator operating at 0.4 m/s. This value is very close to Fruin's computed capacity of 6400 pedestrians per hour (Goodman, 1992).

Since the Fruin study did not list the average behavioral parameters exhibited the pedestrians during the measurement period, the input parameters in this scenario were varied to see how the simulation's computed capacity changed. As it turned out, very similar capacities were found even when changing the speed preferences and lane changing aggressiveness parameters of the various pedestrian classes. It is believed that the reason for this is that under saturated conditions there is insufficient room available on a moving belt system for passing to occur. Moreover, since passing is next to impossible in these conditions it appears that the entire pedestrian stream is limited by the presence of walkers since the users that desire to move faster are unable to achieve these speeds. Therefore, it is believed that the capacity of the belt is tied in large part to the speed at which the belt operates, thereby making the information gathering precautions described above largely unnecessary.

4.1.2.1.2 Capacity Scenario 2

A second scenario was then analyzed to determine the effects of the facial ellipse on the observed capacity. When the model was set to operate in an uphill direction under the same pedestrian parameters with 80% of pedestrians set to observe the facial ellipse, the result was a capacity of 3800 pedestrians per hour. Data presented by Turner in his section of the *Vertical Transportation Handbook* showed a lower limit of capacity for escalators operating at 0.45 meters per second to be 4051 pedestrians per hour (Turner, 1998). Although this simulation lacks the resolution necessary to model a belt speed of 0.45 meters per second, it is believed that the slightly lower capacity found at a slightly lower speed in the simulation matches the findings of the empirical study.

4.1.2.1.3 Capacity Scenario 3

A final scenario was simulated with the passing module disabled to mimic the theoretical study of lane-specific capacity conducted by Davis and Dutta. This study used mathematical approaches to capacity, which fortunately provided ample pedestrian and belt speed data to duplicate their procedures in simulation form. The model computed a belt capacity of 7200 pedestrians per hour, whereas the study found a capacity of 6600 pedestrians per hour (Davis & Dutta, 2002). It is believed that the reason for this discrepancy is that the theoretical study assumed a gap of two treads between walking pedestrians in order to account for climbing motion. Conversely, the simulation determined that at this flow rate the belt would be congested and the users would not be able to sustain a walking pace and instead was able to separate the pedestrians on the belt by a gap of only one tread.

4.1.2.2 *Creation of Time-Space Diagrams*

In order to gain a better understanding of the model's operation, the capability to create time-space diagrams was implemented into the model in Section 3.3.2. As discussed in that section, a time-space diagram is a plot that shows the trajectory of a moving entity with elapsed time plotted on the x-axis and the distance traveled since time zero on the y-axis. This type of plot is beneficial because the interaction between travelers within a system over can be determined from a static plot. Additionally, by inspecting the slope of the various traveler trajectories, the speed of an individual pedestrian can be determined since distance traveled over time elapsed is speed.

Although NetLogo is capable of generating its own plots, these figures lack clarity and are not suitable for detailed analysis. Therefore, the file output module described in Section 3.3.2 was employed to export simulation data for use in this analysis. In order to generate a suitable time-space diagram for analysis, the model was allowed to run for sixty seconds in a low flow environment of 1200 pedestrians per hour under the input parameters present in Example Scenario 0, described in Appendix B, Section B.4. The second-by-second pedestrian data contained in the output file were plotted in Microsoft Excel, and an excerpt from this plot is featured as Figure 4.4, below.

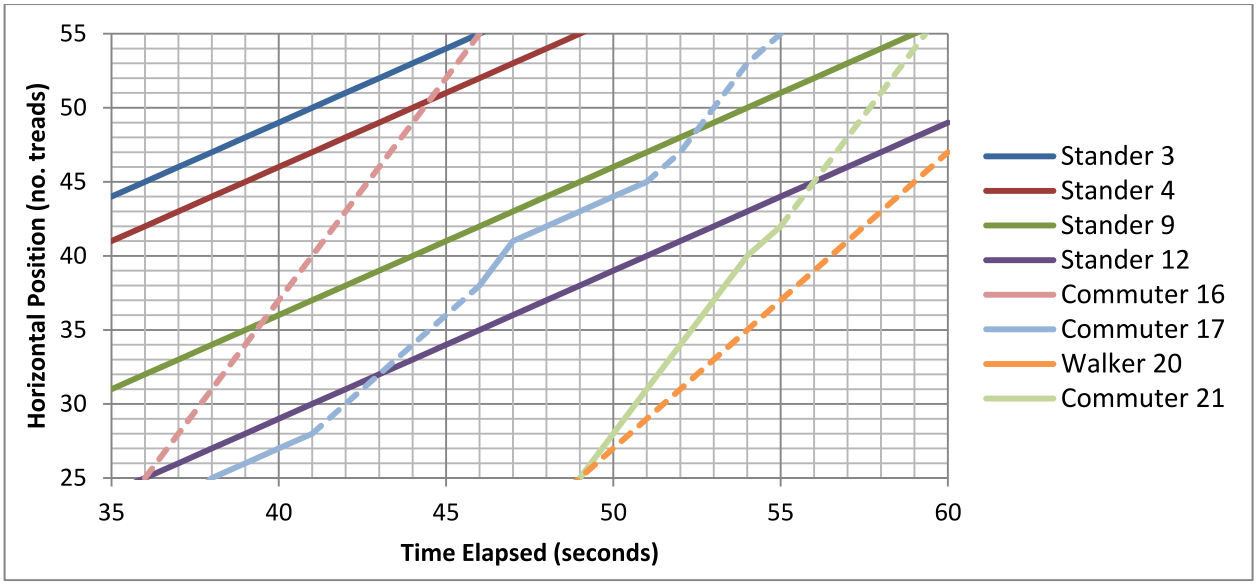


Figure 4.4: Time-Space Diagram Showing Following and Lane Change Behaviors

Figure 4.4 shows the trajectories of eight different simulated travelers observed during a time window of 25 simulation seconds over a 30-tread (12-meter) length of belt. In this figure, the pedestrians are individually color coded, with solid lines representing travel in the right lane and dotted lines indicating travel in the left lane.

The first conclusion that can be drawn from this figure is the distinct speed preferences of each of the three pedestrian classes. The four standers are all observed in this scenario to have a constant slope of 1, indicating that they are traveling at one tread per second (0.4 meters per second), which is the belt speed in Example Scenario 0. Likewise, the orange line representing Walker 20 can be seen to have a constant slope of two treads per second which is equivalent to the belt speed plus its climbing speed of 0.4 meters per second. The three commuters present in this scenario represent a more interesting situation, as their relatively aggressive behavior causes some of them to come into conflict with the other users in the system. While Commuter 16 exhibits a constant speed of three treads per second through a combination of the belt speed and its climbing speed of 0.8 meters per second, this is because it is traveling in the left lane for its entire journey through this section of the belt and therefore does not need to decelerate when it passes the two walkers traveling in the right lane. However, the other two commuters both experience slowdowns when they come upon slower travelers in their current lane and are compelled to take action to maintain their desired speed, as will be discussed below.

Commuter 17 represents perhaps the most interesting case in the time-space diagram showed in Figure 4.4. When it enters this plot at 38 seconds, it is exhibiting a speed of one tread per second, much below its desired speed of three treads per second. This is because it is following two treads behind Stander 12. At time step 41, the dotted line in the figure indicates that Commuter 17 entered the left lane to pass Stander 12, merging back right after completing the pass. For this pedestrian, a "completed pass" is indicated when it has achieved a rear headway of three treads to satisfy the TRANSIMS lane change rules discussed in Section 3.2.3.2. However, once back in the right lane Commuter 17 quickly encounters another stander. Ordinarily, it would immediately switch back left again, but under the parameters of Example Scenario 0 all pedestrians must wait five seconds since their last lane change in order to be eligible to change again. At time step 51, these five seconds have elapsed and the Commuter 17 passes Stander 9, remaining in the left lane for the duration of the time-space diagram.

Commuter 21, on the other hand, appears to be in conflict with Walker 20 when it enters Figure 4.4 at time step 49. However, this behavior can be explained by the fact that Walker 20 is traveling in the left lane with Commuter 21 progressing in the right lane, indicating that Commuter 21 is executing a pass in the right lane. Commuter 21 is able to travel at its full speed of three treads per second until it begins to encounter the effects of Stander 12 at time step 54. However, since Commuter 21 has not changed lanes recently it is immediately able to switch to the left lane and resume its full speed, passing Stander 12 and Stander 9 in the process.

It is in this way that time-space diagrams may be created from the model's output data and used to investigate the performance of the model. Through the examples described above, a brief glimpse into the logic and procedures contained within the model to govern following and lane change choice behaviors can be found. Using observations made of this and other time-space diagrams, the validation of the model can be brought to a successful conclusion.

4.2 Potential Applications

Throughout the process of developing and assessing the performance of the model, the primary focus was on its original purpose of determining the capacity of a belt system through an effective simulation environment. With this goal completed, other potential applications of the model determined during the validation process can be explored.

4.2.1 **Sensitivity Analysis of Belt Systems**

The first and perhaps most obvious application of the model is to perform sensitivity analysis on proposed solutions. Already in Section 4.1.2.1, the capacities computed by the model were subjected to such a sensitivity analysis by varying the input parameters to see what changes resulted in the resulting maximum flows. However, in practical designs it is unlikely that a belt specified by a competent engineer will be forced to operate at capacity in all but perhaps the most extreme peak loading conditions. Instead, by performing sensitivity analysis on facilities at other, less critical load levels it will still be possible to see the belt's performance under a variety of input parameters. By subjecting both proposed and existing facilities to sensitivity analysis, the engineer can gain a better understanding of the suitability of the installation.

Another reason that analyzing the capacity of saturated systems is of less importance than those operating under slightly lower inflows is the fact that at capacity, the behavior of passengers on belt systems makes the capacity of the system inherently less sensitive to changes in the input parameters. At higher flow rates, pedestrians exhibit higher packing efficiencies, meaning that each individual pedestrian has less room available for maneuvering than they ordinarily would. At saturation, the riders are packed so tightly onto the belt that lane changes and even walking becomes impossible. In the absence of these confounding behavioral factors, the capacity of a system is almost exclusively driven by the belt speed. Changes in the input stream in terms of pedestrian mix, choice, and climbing capabilities will make little difference here, except perhaps in the case where luggage carrying can be approximated through use of the facial ellipse feature.

In order to investigate the sensitivity of a moving belt system under nominal conditions, users of the model need only to vary the input stream parameters in ways that could potentially be seen over the life of the installation and track the resulting performance metrics. This particular application would perhaps be most useful in belts with occupancies ranging from 20 to 40 percent of cells utilized, as this density range was identified in the literature as being the range where passing and other choice behaviors have the most impact on the operation of the system (Blue & Adler, 2001), as described in Section 2.1.2.5. Using these techniques, a transit agency could use the model to see what level of inflow stream change causes operational problems for an existing system or exceeds the desired performance metrics for a proposed one.

4.2.2 **Analysis of Proposed Rule Implementation**

Another feature of the model that can be used in belt design is the inclusion of rules that can be implemented on the simulated belt. These rules were included to account for the type of restrictions that the literature indicated were put on belt facilities by various transit operators or potentially by the pedestrians themselves. By activating a rule or combination of rules within the simulation environment, their effect could be noted on the capacity of the belt. Additionally, through observation of the model runs and the use of time-space diagram outputs, the effect of proposed rules on the performance of the belt could be assessed as well.

4.2.2.1 *Pedestrian Behaviors as "Rules"*

The most rudimentary rules contained within the model are the ones that affect how the simulation deals with pedestrian behavior. It can be said that model parameters like the option to enable or disable passing behavior could be considered a "rule" that a transit agency could impose on its patrons. Other quasi-rules that are contained within the model are the ability to cause pedestrians to adhere to strict lane assignment protocol. Conversely, a belt's riders could instead be instructed to ignore their lane assignment if the queue in their lane was longer than that of the adjacent lane by some quantity defined by the user of the model. It is also possible for the user to instruct the pedestrians to ignore their assignment altogether and choose their lane at random and let passing behavior sort out the pedestrians on the belt itself. A further example of this sort of rule set includes the "walk-left/stand-right" behavior that is observed on escalators and moving walks and indeed on highway systems as well.

4.2.2.2 *Stand-Only and Walk-Only Restrictions*

In this section, standing and walking restrictions will be investigated. It was hypothesized that if a transit agency chose to implement a rule on an escalator or moving walkway that forced all pedestrians to remain stationary on the belt, the number of pedestrians that the belt was capable of handling would increase. A potential usage for this scenario may be under emergency evacuation protocols or in the case of severe crush loading on the part of pedestrians trying to exit a transit facility. If it were possible for enough users to disembark from a transit vehicle onto a platform to bring that platform to its capacity, dangerous conditions might occur. Specifically, pedestrians at the edge of the crowd may be thrown off of the platform and onto electrified rails or other hazardous locations as the crowd uncontrollably shifts due to the shockwave behaviors present in high-density crowds (Fruin, 1984).

To address this research question, walk-only and stand-only rules were implemented in the model. Activating the stand-only restriction would cause all pedestrians to become standers whereas activating the walk-only rule would adjust the pedestrian class volumes in order to bump all standers into the walker class. A series of trial runs was then conducted to assess the effect of walking and standing restrictions on the overall capacity of a sample belt through simulation. The scenarios described below can be found in Section B.4 of Appendix B.

4.2.2.2.1 Rule Scenario 1 (Example Scenario 2 in Section B.4)

The first simulation that was conducted to address the research question was treated as a control in that no walking or standing restrictions were enabled on the belt. This scenario took place on an uphill escalator with 95% of passengers observing the facial ellipse using the standard pedestrian mix, speeds, and characteristics that were used in the capacity scenarios featured in Section 4.1.2.1.

The reason an uphill escalator was used in this scenario was to approximate the platform egress situation described above. It was envisioned that this scenario took place in a subterranean transit station, and the only means of escaping the crowded platform was a bank of escalators. Under this particular uphill escalator simulation, the capacity of the belt was found to be 3700 pedestrians per hour.

4.2.2.2.2 Rule Scenario 2 (Example Scenario 4 in Section B.4)

Next, the scenario was repeated except with a rule implemented where users are only permitted to stand on the belt. No walking or climbing is permitted. Although it is uncertain how a transit agency would be able to enforce this restriction, a capacity of 4400 pedestrians per hour was observed, indicating a slight improvement in throughput. Based on observation of the model in this scenario, it was determined that this improvement in capacity was a result of the removal of imperfections in the packing of the belt due to walking. As with any optimization-based situation, the removal of perturbations from the system has the effect of improving the overall performance.

4.2.2.2.3 Rule Scenario 3 (Example Scenario 5 in Section B.4)

Finally, the scenario was run under walk-only rules. With all users climbing the escalator's stairs using either walker or commuter preferences, the belt was observed to exhibit a capacity of 3800 pedestrians per hour. This slight improvement over the baseline scenario was again believed to be a result of the reduction in variability that resulted from the presence of standers in the original case. However, the fact remains that the capacity was lower than that observed in Rule Scenario 2 from packing in standers alone.

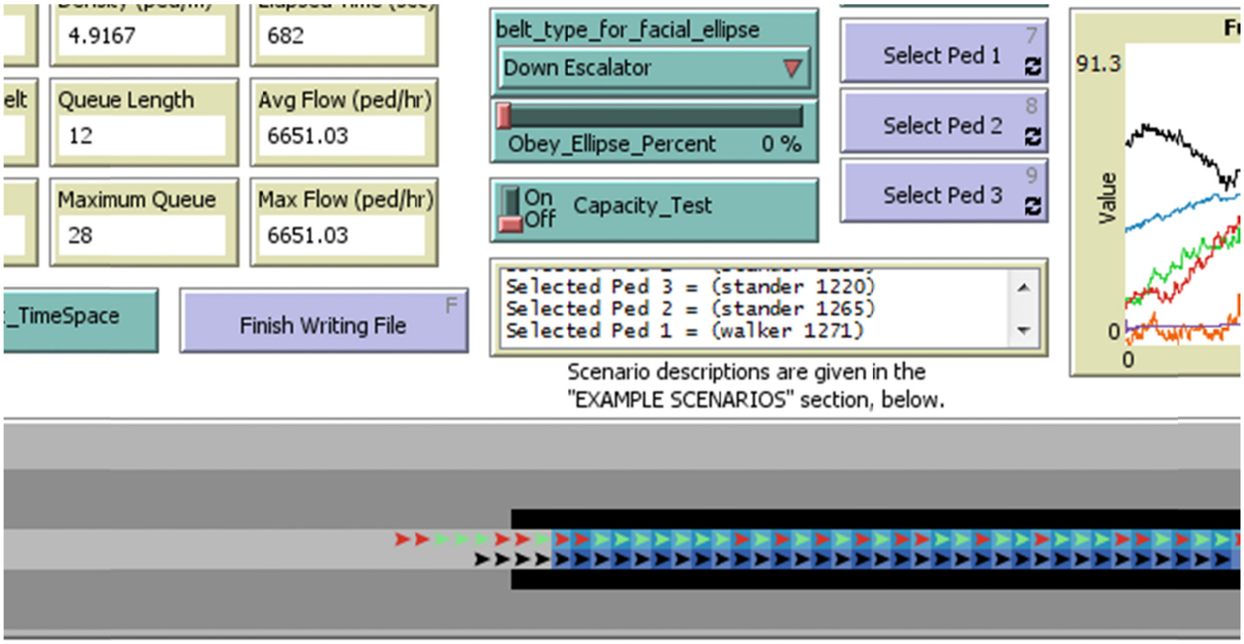
4.2.3 **Platform Sizing**

Based on the analysis performed in Section 4.2.2.2, above, the model can also have a role in platform sizing for transit facilities. As was seen in that section, a moving belt system can have the effect of limiting the rate at which pedestrians can exit a confined area. Since the model tracks the growth of the queue at the base of the belt, as was seen in Figure 3.9, it can also be used to determine the maximum number of pedestrians that will be present in this location under peak hour crush loading conditions.

In addition to the previously discussed scenario where transit patrons are in danger of being shoved off of a transit platform, another situation where it becomes important to assess the queue length that exists at a moving belt facility is when belts are placed in sequence. This sort of situation is found with escalators in deep transit stations and mid-rise office buildings and also with moving walkways in distributed facilities like airports. According to the literature, if designers fail to leave adequate queuing space around escalators and moving walks at major activity centers, the continual driving action of the belt may cause passengers to be conveyed into areas where insufficient space exists to accommodate them, especially under emergency or evacuation situations. This situation presents a public safety hazard since the only way to prevent oneself from being forced into a packed queuing area is to walk backwards along the belt which would require the cooperation of all belt passengers since those riders just entering the belt will likely be unaware of the situation at the downstream end (Mansel, Menaker, & Hartnett, 1998).

Therefore, it is of great importance for engineers and architects to consider platform area and other conditions like queuing area at landings between moving belt facilities in the design of ambulatory facilities. Fortunately, in addition to being able to plot queue behavior over time as was discussed in Section 3.3.3, the model is capable of tracking the maximum queue length recorded in a given simulation. A screenshot of the model as rendered in NetLogo is shown in Figure 4.5, below. In this figure, the instantaneous and maximum queue length meters can be seen towards the upper left, with the current queue present in the model shown in the simulation window near the bottom.

As was seen in the description of the Capacity Test module (Section 3.3.3), at the maximum throughput of a belt facility it can be said that a steady state queue exists. However, it is important to note that a maximum queue length determined in this way is only indicative of one possible outcome under that arrival condition. In order to determine the maximum queue length with confidence, simulations must be run using different random number seeds in order to account for the randomness that is inherent in stochastic arrivals. Using this queue length in conjunction with the amount of space required per pedestrian in order to prevent shockwave behavior from occurring (Section 2.1.1.2), the total amount of area required to handle queuing for a given moving belt system can be determined.


4.9167

682

Queue Length: 12

Avg Flow (ped/hr): 6651.03

Maximum Queue: 28

Max Flow (ped/hr): 6651.03

belt_type_for_facial_ellipse: Down Escalator

Obey_Ellipse_Percent: 0 %

Capacity_Test: On Off

```
Selected Ped 3 = (stander 1220)
Selected Ped 2 = (stander 1265)
Selected Ped 1 = (walker 1271)
```

Scenario descriptions are given in the "EXAMPLE SCENARIOS" section, below.

Figure 4.5: Excerpt of NetLogo Interface Showing Queue Simulation and Tabulation

Finally, it is also important to consider the way that pedestrian flow peaks and ebbs over time in an ambulatory facility. The configuration of the entry to the escalator or moving walkway can have a great impact on the way in which pedestrian arrivals are distributed. Therefore, by designing a transit platform or the entry area of a public space in such a way that arrivals are spread out over time due to varying walking distances, the peak inflow that a moving belt facility must be designed to handle can be reduced, as discussed in Section 2.1.2.3. Additionally, at these higher levels of inflow, the growth of a belt's queue can and should be tracked over time using this module to determine how much storage space is required to hold the queue based on the duration of time that oversaturated conditions are expected to exist.

4.3 Suggestions for Further Research

During the course of this project, three additional research questions presented themselves for which the model was incapable of determining a solution. These principles are put forth here in the hopes that another, future project will be able to address them.

4.3.1 **Belt Performance in Terms of Delay**

The performance metrics used by this model are almost exclusively designed for use in capacity analysis and in ensuring that the fundamental characteristics of this traffic flow system can be monitored. To that end, the model features real-time displays and file output protocols to present information about statistics like instantaneous flow, speed, and density. The maximum and average values of flow throughout the simulation period can be used to draw conclusions about the capacity, while additional displays that show queue length measurements can help the engineer to make decisions about the design and sizing of platforms and landing areas between belts.

However, another performance metric that may prove useful in conducting analyses of moving belt systems is that of delay. Delay is a common measure of effectiveness for other transportation facilities, allowing a designer to determine what level of impedance a user will experience when passing through an intersection or transit station. Therefore, it is proposed that the model be adjusted to compute the delay incurred by each pedestrian as they traverse the modeled moving belt system. This particular value could be computed as a difference between the travel time and the amount of time that they would have taken walking at their desired speed. Separate values could be computed to determine delay incurred on the belt and in the queue.

The resulting delay measures for the entire system could prove to be another important performance characteristic for the engineer to consider when designing an ambulatory facility. The inclusion of this performance metric would represent a significant shift in the approach taken by engineers in the design of moving belt systems, as up until this point the primary concern in the design of these systems has been capacity. This paper has also previously proposed designing moving belt systems to achieve a density of no more than one passenger per square meter – an occupancy of 40% – in order to allow pedestrian choice behavior on the belt during normal periods. Accounting for pedestrian delay in a system in addition to these two measures of effectiveness could provide a better understanding of the proposed solutions to the problem of pedestrian locomotion.

Further research will need to be conducted before this proposal can be implemented. Most notably, study is merited to determine whether the computed delay present on a moving belt will even be significant enough to track. The question for this study to answer will be to find how much delay is incurred by a single pedestrian on a congested belt and whether this amount is enough to cause a change in behavior. It is entirely possible that pedestrians will be unconcerned with a delay of fifteen or twenty seconds compared to traveling on an uncongested belt at their desired speed so long as the capacity exists on the belt system for them to reach their destination. If further study shows to indeed be the case, then it may be that the inclusion of delay computation is not a significant factor to consider in belt design relative to the issues of capacity and occupancy.

4.3.2 Microsimulation of Bottleneck Effects

Although the bottleneck effect utilized in this model appears to give a good approximation of the level of constriction exhibited in actual escalator and moving walkway systems, the fact remains that this effect is largely serendipitous. Therefore, it is proposed that a future project be undertaken to attempt to model the bottleneck effect using microsimulation.

In order to perform this analysis, a continuous simulation of pedestrian behavior would need to be created. Pedestrians could be generated some distance away from the belt at randomly determined positions around the entry constriction. Then, using pedestrian wayfinding techniques, the travelers would make their way onto the belt. The NetLogo software would lend itself well to this application, as the program is capable of performing the kind of continuous simulation required here in addition to the cellular automata approach used in the model above. Further detail regarding this proposal can be found in the model documentation in Section B.6 of Appendix B.

The most notable challenge in completing this research is the difficulty in modeling pedestrian wayfinding in crowd scenarios. Hoogendoorn's experiments included some empirical studies of crowds funneling themselves down into bottlenecks, including some fascinating spatial trajectory figures showing dynamic queue formation (Hoogendoorn & Daamen, 2005). Unfortunately, these studies were limited to observationally defining queue characteristics and lack definition on the methods by which the pedestrians made their way into the queue lanes.

4.3.3 Effect of Impatience on Pedestrian Aggressiveness

Observations made in validating the model's passing behavior in Section 4.1.1.2 lead to another suggestion for future research. It was hypothesized that as the amount of time a pedestrian is stuck following a slower traveler ahead of them, the more impatient they will be come. It is believed that in this situation, impatience would manifest itself through more aggressive behavior like the relaxation of choice parameters such as the amount of free space back that is required to complete a merging action. If the hypothesis proves to be true, then after some amount of time that a pedestrian spends traveling below their desired speed the value of their lane changing acceptance factors would become more lenient.

However, the inclusion of this function into the model would likely require original empirical study since minimal information about the growth of pedestrian impatience with time was found during the literature review for this project.

Chapter 5 – Conclusions

The goal of this project was to develop a theoretical framework to model pedestrian behavior on moving belt facilities like escalators and moving walkways using microsimulation. By drawing information about pedestrian capabilities and choice behavior from a number of sources in the literature – including studies that were originally intended to apply only to the automotive environment – a comprehensive and logical basis for the model was created. These facts, practices, and principles were then coded in an object-oriented programming and modeling software called NetLogo to create a simulation environment and the controls necessary to operate it. Finally, a series of tests were run to validate the performance of the model and to gain a better understanding of how it could be utilized to answer questions about the capacity, sensitivity, and queue growth of moving belt facilities.

5.1 *Theoretical Approach and Modeling Strategy*

At the core of this research was the desire to create a model of pedestrian behavior on continuously moving belts like escalators and moving walkways. At present, these facilities are designed and sized using capacity curves that are based on empirical studies of isolated escalator installations, as was discussed in Section 1.3.2. While adjustment factors are often used to convert these theoretical capacities into practical ones, the fact remains that capacity curves only give the system's condition under a single set of inflow parameters. Examples of theoretical and practical capacity curves from a manufacturer's design literature can be seen as the red and blue lines, respectively, in Figure 5.1, below. Furthermore, flow information is only provided at capacity. As was seen in Section 4.2.1, the performance of a moving belt system at lower occupancy levels is at least as important to system designers as how it behaves at capacity given that these lower densities will provide a more desirable state of operation in terms of pedestrian choice and comfort.

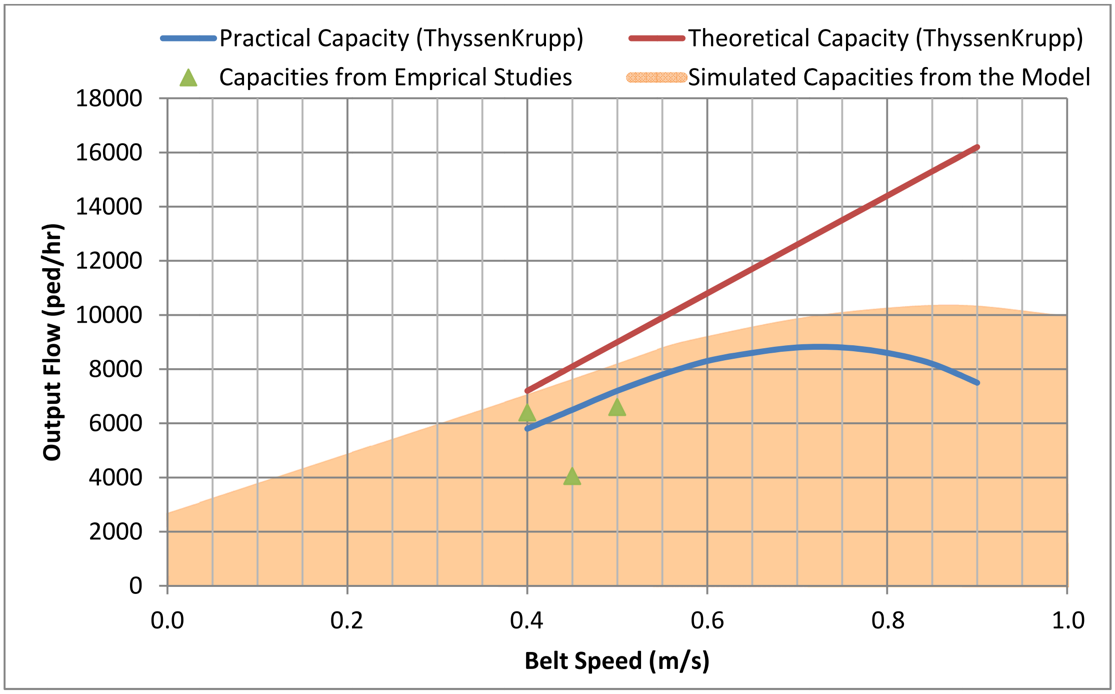


Figure 5.1: Comparison of Escalator Capacities from the Model and from the Literature

In order to combat these deficiencies in the state of the practice, a model was constructed to create a microsimulation of pedestrian choice behavior on moving belt facilities. By creating a simulation of pedestrian behavior, an experimental approach towards moving belt research could be utilized. In this way, not only could the performance of an escalator or moving walkway installation be tested at varying levels of occupancy up to its capacity, different inflow conditions could also be tested to see the effects of changes in the pedestrian stream on belt performance. Experimental changes to be implemented in the model included input flow, speed distribution, user aggressiveness, belt characteristics, and even passing and walking restrictions that could be implemented by an operator like a transit agency.

However, the functionality of the proposed model as a research tool relied on its ability to mimic real-world behaviors. While ample research existed on pedestrian characteristics like walking speed and stair-climbing ability as well as moving belt specifications like speed, very little information could be found about pedestrian choice behavior in confined spaces like those present on escalators and moving walkways. Most research into the following and passing choices made by pedestrians dealt with wayfinding and navigation through crowds in ambulatory facilities like airports and shopping centers, not through what effectively amounted to a walled-in corridor. To combat this shortcoming in the literature, it was decided that this model would use findings from another field of transportation engineering as an analogue for the decisions made by travelers on escalators and moving walkways.

Through the initial literature review as well as everyday observation, it was found that pedestrians on belts automatically segregate themselves into two layers, one for slower traffic and one for passing. These parallel layers mimic the lanes found on a typical road, including the following and passing choice behaviors exhibited by drivers. Therefore, it was determined that automotive research, specifically the field of car-following behavior, would be the source of the choice behaviors programmed into the model. Specifically, the behavioral rules found in the microsimulator module of TRANSIMS were implemented into the model because of the way that rule-based behavioral models are especially suited to governing pedestrian choice behavior. Further benefits were gained because of the similarity between the standardized spacing of step treads on an escalator and the discrete location information found in the cellular automata simulation framework that comprises the TRANSIMS microsimulator.

5.2 Comparison to Empirical Pedestrian Behavior and Capacity Data

After the model had been successfully programmed in the cellular automata microsimulation environment, it had to be validated to ensure that it matched closely with real-world behaviors. In addition to simple visual verification that the programmed behavioral rules were followed, time-space diagrams were also utilized to create a more permanent record of passing and following choice. Throughout these processes, it was determined that the simulation accurately modeled pedestrian behavior in that each of the simulated entities within the system moved in accordance with observed real-world pedestrian behavior and that the simulated users interacted in much the same way as their physical counterparts. Furthermore, the microsimulated pedestrians each possessed internal variables that accurately accounted for their own individual desires and performance.

However, the basic research questions of this study were answered through an analysis of the model's outputs. Data can be collected both on a pedestrian-by-pedestrian basis as well as on a system-wide level for all three fundamental characteristics, specifically speed, flow, and density. Not only are the instantaneous, average, and maximum values of these characteristics tracked, other useful parameters like queue length for the system and headway for the users are also recorded. However, the advantage of a simulation of pedestrian flow over empirical studies is that changes in the operational state of the system can be modeled. As was stated previously, the goal of this project was to model both changes in rules implemented on a moving belt system by the operator as well as variations that may occur in the pedestrian stream entering an escalator or moving walkway.

To this end, several capacity tests were undertaken to compare the results of the model to empirically determined capacities found in the literature in order to continue with the validation of the model as a tool for both research and design. In Section 4.1.2.1, several studies from design manuals and research studies were simulated using the model, with the resulting capacity from the model closely matching those found in the literature for a variety of operational scenarios.

However, the real power of the model comes from its ability to simulate belt operations under a variety of conditions. The previous state of the practice of belt specification was to rely on capacity curves, as was previously seen above in Figure 5.1, or empirical studies like those that were used in the capacity tests, shown as green triangles in the same figure. In contrast, this model can simulate a limitless number of scenarios over a range of belt speeds and inflow conditions, resulting in the orange region of output capacities seen in Figure 5.1 that cover the variety of conditions that can be found in belt installations around the world. Perhaps the most important aspect of the model is that it can evaluate the performance of a belt in unsaturated conditions to ensure that it remains at a sufficiently low occupancy to ensure the comfort of its riders through their ability to make their own behavioral choices, giving the wide range of output flows seen in the above figure.

5.3 Real-World Applications of the Model

Obviously, a microsimulated model of pedestrian behavior can be used to experimentally predict the characteristics and performance of the physical system it is meant to represent. It has previously been seen that capacity of a system can be found under a user-defined set of inflow parameters. However, there are several additional applications to which the model may be suited that were discussed in Section 4.2.

The first potential application is not particular dissimilar to the model's primary purpose of capacity determination, that of performing sensitivity analyses of belt systems. By assessing the performance of a proposed system under a range of inflow stream settings, uncertainty in demand prediction can be accounted for in the design process. However, this same analysis can be applied to existing systems by varying the input stream and examining the result these changes have on the operation of a moving belt system in order to determine how long an existing facility can be expected to operate under nominal conditions before improvement or replacement becomes necessary.

As was discussed above, the model can determine the capacity of a moving belt facility under the expected set of input characteristics as well as to test the sensitivity of this sort of solution to changes in that input. However, other experimental tests can also be run using the simulation framework. One such example described above involves analyzing the performance of an escalator under a proposed set of operational rules that could potentially be implemented by a transit authority or other operator, specifically a rule that all users must stand still on the belt was examined. Under emergency conditions, it may become necessary to direct users to pack more efficiently onto an escalator in an attempt to increase the overall flow rate leaving a subterranean transit facility, for instance. An experiment conducted in Section 4.2.2.2 appears to indicate that this sort of rule has the potential of increasing pedestrian throughput by a significant margin, on the order of 15 to 20 percent.

Finally, because of its ability to track queue length over the course of a simulation, the model seems to be suited for use as a tool for platform sizing. Given the stochastic nature of arrivals in a conventional belt installation or even the ebb and flow seen in transit facilities as trains and buses arrive and discharge their passenger loads, there exists the potential for significant queue accumulation at the entry to an escalator or moving walkway. By tracking the length of this queue over time and accounting for the minimum amount of space required per person in order to ensure pedestrian comfort and control and to prevent the formation of shockwaves in a crowd, it becomes possible to determine just how much space is required to safely contain the expected queue. If this area is supplied in the design of the waiting area at the entry to the moving belt, whether on a transit platform or simply at a landing between moving walkways, the safety of a system's users can be preserved.

5.4 Summary of Contributions of the Project

Based on the results presented in this paper, it seems that the fundamental approach of quantifying and modeling pedestrian behavior on a moving belt system using microsimulation is sound. Furthermore, the method by which the model was constructed – using data from a number of different disciplines within the field of transportation engineering – appears to provide a satisfactory framework for pedestrian behavior. In addition, it has been shown that the model created in this project can not only be used to analyze the capacity and fundamental characteristics of a belt system, it can also fulfill other functions such as providing plots of pedestrian trajectory and assessing the effect of rule implementation on the operation of the system under both normal conditions or under evacuation protocols.

In this way, it seems that the model provides a new approach to the design of pedestrian facilities like escalators and moving walkways. By allowing the user to input the specific parameters exhibited by pedestrians in that region, the engineer's ability to accurately predict moving belt performance is greatly improved. In addition to the benefits to capacity analysis, significant gains are also observed in how well the growth of a queue and its behavior over time can be anticipated. Therefore, it is hoped that this model, or at least the approach it takes to quantifying and modeling pedestrian behavior, is implemented in the field of moving belt specification as a tool for both research and design.

References

Ames, N. (1859). *Patent No. 25076.* United States of America.

Bangash, M. H., & Bangash, T. (2007). *Lifts, Elevators, Escalators and Moving Walkways/Travelators.* New York: Taylor & Francis.

Blue, V. J., & Adler, J. L. (1998). Emergent fundamental pedestrian flows from cellular automata microsimulation. *Transportation Research Record, 1644*, 29-36.

Blue, V. J., & Adler, J. L. (2001). Cellular automata microsimulation for modeling bi-directional pedestrian walkways. *Transportation Research Part B*, 293-312.

Cepolina, E., & Tyler, N. (2005). Understanding capacity drop for designing pedestrian environments. *Walk21-VI "Everyday Walking Culture", The 6th International Conference on Walking in the 21st Century.* Zurich: University College London.

Daamen, W., & Hoogendoorn, S. P. (2003). Experimental Research of Pedestrian Walking Behavior. *Transportation Research Record, 1828*, 20-30.

Davis, P., & Dutta, G. (2002). *Estimation of Capacity of Escalators in London Underground.* London: London School of Economics and Political Sciences.

Donoghue, E. A. (1981). *Handbook on A17.1: Safety Code for Elevators and Escalators.* New York: American Society of Mechanical Engineers.

Fruin, J. J. (1984, May). Crowd Dynamics and Auditorium Managment. *Auditorium News*.

Fujiyama, T., & Tyler, N. (2004). *An Explicit Study on Walking Speeds of Pedestrians on Stairs.* London: Centre for Transport Studies, University College London.

Geismar, T., Chwast, S., de Harak, R., Lees, J., & Vignelli, M. (2011). *Symbol Signs*. Retrieved January 1, 2011, from AIGA | the professional association for design: http://www.aiga.org/content.cfm/symbol-signs

Goodman, L. (1992). Transportation Interface Areas. In J. D. Edwards, & I. o. Engineers (Ed.), *Transportation Planning Handbook* (pp. 201-293). Englewood Cliffs, New Jersey: Prentice-Hall.

Hession, A. (n.d.). *NEII On-Line Newsroom*. Retrieved October 12, 2010, from National Elevator Industry, Inc.: http://www.neii.org/presskit/pressmaster.cfm?link=7

Hobeika, A. G., & Gu, Y. (2004). TRANSIMS - the next generation planning/simulation model. *Urban Transport X*, 141-151.

Hoogendoorn, S. (2004). Walking behavior in bottlenecks and its implications for capacity. *TRB 2004 Annual Meeting CD-ROM.* Washington: Transportation Research Board of the National Academies.

Hoogendoorn, S. P., & Daamen, W. (2005). Pedestrian Behavior at Bottlenecks. *Transportation Science, 39*(2), 147-159.

Hoogendoorn, S., & Bovy, P. H. (2000). Gas-Kinetic Modeling and Simulation of Pedestrian Flows. *Transportation Research Record, 1710*, 28-36.

Hoogendoorn, S., & Bovy, P. H. (2003). Simulation of pedestrian flows by optimal control and differential games. *Optimal Control Applications and Methods, 24*(3), 153-172.

Lampugnani, V. M., Hartwig, L., Simmen, J., & Imorde, J. (1994). *Vertical: Lift, Escalator, Paternoster.* Berlin: Ernst & Sohn.

Lee, Y.-C. (2005). *Pedestrian Walking and Choice Behavior on Stairways and Escalators in Public Transport Facilities.*

Los Alamos National Laboratory. (2006). *TRansportation ANalysis SIMulation System (TRANSIMS) Course Manual: Microsimulator.* AECOM.

Mansel, D. M., Menaker, P. J., & Hartnett, G. (1998). High-Capacity Light Rail Transit: Balancing Stationside and Railside Capacities. *Transportation Research Record, 1623*, 170-178.

May, A. D. (1990). *Traffic Flow Fundamentals.* Englewood Cliffs, NJ: Prentice Hall.

O'Neill, R. (1974). Escalators in Rapid Transit Stations. *American Society of Civil Engineers Jornal of the Transportation Engineering Division*, 1-12.

Rakha, H., & Crowther, B. (2002). A Comparison of the Greenshields, Pipes, and Van Aerde Car-Following and Traffic Stream Models. *Transportation Research Record, 1802*, 248-262.

Rakha, H., Pecker, C. C., & Cybis, H. B. (2007). Calibration Procedure for Gipps Car-Following Model. *Transportation Research Record, 1999*, 115-127.

Rickert, M., Nagel, K., Schreckenberg, M., & Latour, A. (1996). Two lane traffic simulations using cellular automata. *Physica A: Statistical and Theoretical Physics, 231*(4), 534-550.

Schindler Escalators. (2010). *Schindler 9300 Advanced Edition.* Morristown, New Jersey: Schindler USA.

Slaughter, L. (2004). *108 Congress HR 4995.*

Sutherland, D. H., Olshen, R. A., Biden, E. N., & Wyatt, M. P. (1988). *The Development of Mature Walking.* Philadelphia: J.B. Lippincott.

Sutthawassuntorn, K. (2010). *Study and Analysis of Passenger Movement and Service on Escalators at Metro Station.*

ThyssenKrupp Elevator. (2004). *Escalators and Moving Walks planning guide.* Memphis, Tennessee: ThyssenKrupp Americas Business Unit.

Turner, D. L. (1998). Escalators and Moving Walks. In G. R. Strakosch, *The Vertical Transportation Handbook, Third Edition* (pp. 210-236). New York: Wiley.

Welch, P., Houseman, H., Kinihan, J., Evans, D., Herndobler, R., Nurnberg, T., et al. (2009). Heavy Duty Transportation System Escalator Design Guidelines. In *APTA Rail Transit Standards, Volume 5: Fixed Structures* (pp. 7.1-7.26). Washington: American Public Transportation Association.

Wilensky, U., & Tisue, S. (2004). NetLogo: A Simple Environment for Modeling Complexity. *International Conference on Complex Systems.* Boston.

WMATA. (2008). *Station Site and Access Planning Manual.* Washington: Washington Metropolitan Area Transit Authority.

WMATA. (2008). *Washington Metro Station Access and Capacity Study.* Washington: Washington Metropolitan Area Transit Authority.

Appendix A – NetLogo Model Code

```
;; A.1  Create Global Variables
globals
[
 treads
 inc_create_walk
 inc_create_stand
 inc_create_comm
 v_belt
 v_walk
 v_comm
 p_stand
 p_walk
 p_comm
 peds_in
 peds_in_belt
 standers_in
 walkers_in
 commuters_in
 peds_out
 standers_out
 walkers_out
 commuters_out
 flow
 flow_max
 density
 n_stand
 n_walk
 n_comm
 o_ellipse
 p_ellipse
 p_decel
 avg_speed_stand
 avg_speed_walk
 avg_speed_comm
 queue
 queue_right
 queue_left
 queue_removed
 queue_total
 max_queue
 time
 selected1
 selected2
 selected3
]

;; A.2  Create Turtle-Specific Variables
turtles-own
[
  speed
  speed_desired
  speed_desired_now
  rand_decel
  distance_forward
  gap_forward
```

```
  gap_back
  gap_forward_other
  gap_back_other
  tolerable_gap_back_other
  speed_desired_now_back
  speed_desired_now_back_other
  speed_desired_now_forward
  speed_desired_now_forward_other
  time_since_lane_change
  facial_ellipse
  lane
  on_belt
]

;; A.3  Create Pedestrian Classes
breed [ standers stander ]
breed [ walkers walker ]
breed [ commuters commuter ]

;; A.4  Define Example Scenarios
to scenario0
  ca
  set belt_length 24
  set Stand_Percent 25
  set FastWalker_Percent 60
  set beltspeed 0.4
  set pedspeed_stander 0.4
  set pedspeed_walker 0.4
  set pedspeed_commuter 0.8
  set climbspeed_walker 0.4
  set climbspeed_commuter 0.8
  set lanechange_allowed true
  set slow_with_distance false
  set belt_rules "None"
  set belt_type_for_facial_ellipse "Down Escalator"
  set obey_ellipse_percent 0
  set initially_populate_belt true
  set mergespace_walker 0.8
  set mergespace_commuter 0.4
  set Max_Queue_Diff 20
  set ignore_lane_assign true
  set inflow 2500
  set stopafter_seconds 3600
  set decel_probability 3
  set min_t_btwn_lanechange 5
  set capacity_test false
  output-print "Scenario 0 - TNV D 2500 0.4"
  set File_Output_TimeSpace false
end

to scenario1
  ca
  set belt_length 24
  set Stand_Percent 50
  set FastWalker_Percent 50
  set beltspeed 0.4
  set pedspeed_stander 0.4
```

```
  set pedspeed_walker 0.4
  set pedspeed_commuter 0.8
  set climbspeed_walker 0.4
  set climbspeed_commuter 0.8
  set lanechange_allowed false
  set slow_with_distance false
  set belt_rules "None"
  set belt_type_for_facial_ellipse "Down Escalator"
  set obey_ellipse_percent 0
  set initially_populate_belt true
  set mergespace_walker 0.8
  set mergespace_commuter 0.4
  set Max_Queue_Diff 20
  set ignore_lane_assign false
  set inflow 3500
  set stopafter_seconds 3600
  set decel_probability 3
  set min_t_btwn_lanechange 5
  set capacity_test false
  output-print "Scenario 1 - FNX D 3500 0.4"
  set File_Output_TimeSpace false
end

to scenario2
  ca
  set belt_length 24
  set Stand_Percent 50
  set FastWalker_Percent 50
  set beltspeed 0.4
  set pedspeed_stander 0.4
  set pedspeed_walker 0.4
  set pedspeed_commuter 0.8
  set climbspeed_walker 0.4
  set climbspeed_commuter 0.8
  set lanechange_allowed false
  set slow_with_distance false
  set belt_rules "None"
  set belt_type_for_facial_ellipse "Up Escalator"
  set obey_ellipse_percent 95
  set initially_populate_belt true
  set mergespace_walker 0.8
  set mergespace_commuter 0.4
  set Max_Queue_Diff 20
  set ignore_lane_assign false
  set inflow 3500
  set stopafter_seconds 3600
  set decel_probability 3
  set min_t_btwn_lanechange 5
  set capacity_test false
  output-print "Scenario 2 - FNX U-95 3500 0.4"
  set File_Output_TimeSpace false
end

to scenario3
  ca
  set belt_length 24
  set Stand_Percent 50
```

```
  set FastWalker_Percent 50
  set beltspeed 0.8
  set pedspeed_stander 0.4
  set pedspeed_walker 0.4
  set pedspeed_commuter 0.8
  set climbspeed_walker 0.4
  set climbspeed_commuter 0.8
  set lanechange_allowed false
  set slow_with_distance false
  set belt_rules "None"
  set belt_type_for_facial_ellipse "Up Escalator"
  set obey_ellipse_percent 95
  set initially_populate_belt true
  set mergespace_walker 0.8
  set mergespace_commuter 0.4
  set Max_Queue_Diff 20
  set ignore_lane_assign false
  set inflow 3500
  set stopafter_seconds 3600
  set decel_probability 3
  set min_t_btwn_lanechange 5
  set capacity_test false
  output-print "Scenario 3 - FNX U-95 3500 0.8"
  set File_Output_TimeSpace false
end

to scenario4
  ca
  set belt_length 24
  set Stand_Percent 100
  set FastWalker_Percent 50
  set beltspeed 0.4
  set pedspeed_stander 0.4
  set pedspeed_walker 0.4
  set pedspeed_commuter 0.8
  set climbspeed_walker 0.4
  set climbspeed_commuter 0.8
  set lanechange_allowed false
  set slow_with_distance false
  set belt_rules "Stand Only"
  set belt_type_for_facial_ellipse "Up Escalator"
  set obey_ellipse_percent 95
  set initially_populate_belt true
  set mergespace_walker 0.8
  set mergespace_commuter 0.4
  set Max_Queue_Diff 20
  set ignore_lane_assign false
  set inflow 3500
  set stopafter_seconds 3600
  set decel_probability 3
  set min_t_btwn_lanechange 5
  set capacity_test false
  output-print "Scenario 4 - FSX U-95 3500 0.4"
  set File_Output_TimeSpace false
end

to scenario5
```

```
  ca
  set belt_length 24
  set Stand_Percent 0
  set FastWalker_Percent 50
  set beltspeed 0.4
  set pedspeed_stander 0.4
  set pedspeed_walker 0.4
  set pedspeed_commuter 0.8
  set climbspeed_walker 0.4
  set climbspeed_commuter 0.8
  set lanechange_allowed false
  set slow_with_distance false
  set belt_rules "Walk Only"
  set belt_type_for_facial_ellipse "Up Escalator"
  set obey_ellipse_percent 95
  set initially_populate_belt true
  set mergespace_walker 0.8
  set mergespace_commuter 0.4
  set Max_Queue_Diff 20
  set ignore_lane_assign false
  set inflow 3500
  set stopafter_seconds 3600
  set decel_probability 3
  set min_t_btwn_lanechange 5
  set capacity_test false
  output-print "Scenario 5 - FWX U-95 3500 0.4"
  set File_Output_TimeSpace false
end

to scenario6
  ca
  set belt_length 24
  set Stand_Percent 50
  set FastWalker_Percent 50
  set beltspeed 0.4
  set pedspeed_stander 0.4
  set pedspeed_walker 0.4
  set pedspeed_commuter 0.8
  set climbspeed_walker 0.4
  set climbspeed_commuter 0.8
  set lanechange_allowed false
  set slow_with_distance true
  set belt_rules "None"
  set belt_type_for_facial_ellipse "Up Escalator"
  set obey_ellipse_percent 0
  set initially_populate_belt true
  set mergespace_walker 0.8
  set mergespace_commuter 0.4
  set Max_Queue_Diff 20
  set ignore_lane_assign false
  set inflow 3500
  set stopafter_seconds 3600
  set decel_probability 3
  set min_t_btwn_lanechange 5
  set capacity_test false
  output-print "Scenario 6 - FNX U-0 3500 0.4 SLOW"
  set File_Output_TimeSpace false
```

```
end

;; A.5  Select Pedestrians for NetLogo Time-Space Diagram
to select1
  if mouse-down?
  [
    let mx mouse-xcor
    let my mouse-ycor
    if any? turtles-on patch mx my
    [
      ask selected1
      [
        if (breed = standers) [ set color black ]
        if (breed = walkers) [ set color lime ]
        if (breed = commuters) [ set color red ]
      ]
      set selected1 one-of turtles-on patch mx my
      output-type "Selected Ped 1 = " output-print selected1
      ask selected1 [ set color orange ]
      display
    ]
  ]
end

to select2
  if mouse-down?
  [
    let mx mouse-xcor
    let my mouse-ycor
    if any? turtles-on patch mx my
    [
      ask selected2
      [
        if (breed = standers) [ set color black ]
        if (breed = walkers) [ set color lime ]
        if (breed = commuters) [ set color red ]
      ]
      set selected2 one-of turtles-on patch mx my
      output-type "Selected Ped 2 = " output-print selected2
      ask selected2 [ set color yellow ]
      display
    ]
  ]
end

to select3
  if mouse-down?
  [
    let mx mouse-xcor
    let my mouse-ycor
    if any? turtles-on patch mx my
    [
      ask selected3
      [
        if (breed = standers) [ set color black ]
        if (breed = walkers) [ set color lime ]
        if (breed = commuters) [ set color red ]
```

```
      ]
      set selected3 one-of turtles-on patch mx my
      output-type "Selected Ped 3 = " output-print selected3
      ask selected3 [ set color magenta ]
      display
    ]
  ]
end

;; A.6  SETUP routine
to setup
  clear-all
  constraints
  compute-variables
  draw-belt
  if (initially_populate_belt = true) [populate-belt]
  if (file_output_timespace = true ) [setup-file]
end

;; A.7  Adjust for Constraints module
to constraints
  if beltspeed = 0
  [ set belt_rules "Walk Only" ]
  if belt_rules = "Stand Only"
  [
    set Stand_Percent 100
    set lanechange_allowed false
  ]
  if belt_rules = "Walk Only"
  [
    ;; assume all standers become normal walkers
    set FastWalker_Percent ((1 - (Stand_Percent / 100)) * FastWalker_Percent)
    set Stand_Percent 0
  ]
  ifelse belt_type_for_facial_ellipse = "Down Escalator"
  [
    set o_ellipse false
    set p_ellipse 0
  ]
  [
    set o_ellipse true
    set p_ellipse (obey_ellipse_percent / 100)
  ]
  if slow_with_distance = true
  [ set belt_type_for_facial_ellipse "Up Escalator" ]
end

;; A.8  Compute Variables module
to compute-variables
  set treads (Belt_Length * 2.5) ;; convert meters to number of treads
  set v_belt (BeltSpeed / 0.4)  ;; convert speeds in m/s to cells/tick
  set v_walk (ClimbSpeed_Walker / 0.4)
  set v_comm (ClimbSpeed_Commuter / 0.4)
  set p_stand (PedSpeed_Stander / 0.4)
  set p_walk (PedSpeed_Walker / 0.4)
  set p_comm (PedSpeed_Commuter / 0.4)
  set n_stand (Stand_Percent / 100 * inflow)
```

```
  set n_walk (((100 - Stand_Percent) / 100) * ((100 - FastWalker_Percent) /
100) * inflow)
  set n_comm (((100 - Stand_Percent) / 100) * (FastWalker_Percent / 100) *
inflow)
  set standers_in 1
  set walkers_in 1
  set commuters_in 1
  set p_decel (decel_probability / 100)
end

;; A.9  Draw Simulation Environment module
to draw-belt
  ask patches
  [
    set pcolor gray
    if ((pycor > -2) and (pycor < 3) and (pxcor < treads + 2) and (pxcor > -
3))
    [ set pcolor black ]
    if ((pycor > -1) and (pycor < 2))
    [ set pcolor gray + 2 ]
    if ((pycor = 0) and (pxcor > -1) and (pxcor < treads))
    [ set pcolor blue]
    if ((pycor = 1) and (pxcor > -1) and (pxcor < treads))
    [ set pcolor sky]
    if ((pycor = 0) and (pxcor > -1) and (pxcor < treads) and ((pxcor mod 2)
= 0))
    [ set pcolor blue + 1]
    if ((pycor = 1) and (pxcor > -1) and (pxcor < treads) and ((pxcor mod 2)
= 0))
    [ set pcolor sky + 1]
  ]
end

;; A.10 Populate Belt module
to populate-belt
  if (belt_rules != "Walk Only") [
  set inc_create_stand random-poisson (n_stand / 3600 * (treads / (v_belt)))
  create-standers inc_create_stand
  [
    set color black
    ifelse (belt_rules = "None")
    [ set lane 0]
    [ set lane (random 2)]
    setxy (random treads) lane
    set heading 90
    set speed_desired 0
    set time_since_lane_change 0
    set speed_desired_now speed_desired
    set speed (speed_desired_now + v_belt)
    set on_belt true
    set time_since_lane_change Min_t_btwn_lanechange
    ifelse o_ellipse = true
    [ ifelse (p_ellipse >= (random-float 1))
      [ set facial_ellipse 1 ]
      [ set facial_ellipse 0 ] ]
    [ set facial_ellipse 0 ]
  ] ]
```

```
  set inc_create_walk random-poisson (n_walk / 3600 * (treads / (v_belt +
v_walk)))
  create-walkers inc_create_walk
  [
    set color (lime + 2)
    ifelse (lanechange_allowed = false)
    [ ifelse (ignore_lane_assign = false)
      [ ifelse (belt_rules = "None") [ set lane 1 ] [ set lane 0 ] ]
      [ set lane (random 2) ] ]
    [ ifelse ((belt_rules = "None") and (ignore_lane_assign = false))
      [ set lane 1 ]
      [ set lane (random 2) ] ]
    setxy (random treads) lane
    set heading 90
    set time_since_lane_change 0
    set speed_desired v_walk
    set speed_desired_now speed_desired
    set speed (speed_desired_now + v_belt)
    set tolerable_gap_back_other (mergespace_walker / 0.4)
    set on_belt true
    set time_since_lane_change Min_t_btwn_lanechange
    ifelse o_ellipse = true
    [ ifelse (p_ellipse >= (random-float 1))
      [ set facial_ellipse 1 ]
      [ set facial_ellipse 0 ] ]
    [ set facial_ellipse 0 ]
  ]
  set inc_create_comm random-poisson (n_comm / 3600 * (treads / (v_belt +
v_comm)))
  create-commuters inc_create_comm
  [
    set color red
    ifelse (lanechange_allowed = false)
    [ ifelse (ignore_lane_assign = false)
      [ set lane 1 ]
      [ set lane (random 2) ] ]
    [ ifelse ((belt_rules = "None") and (ignore_lane_assign = false))
      [ set lane 1 ]
      [ set lane (random 2) ] ]
    setxy (random treads) lane
    set heading 90
    set time_since_lane_change 0
    set speed_desired v_comm
    set speed_desired_now speed_desired
    set speed (speed_desired_now + v_belt)
    set tolerable_gap_back_other (mergespace_commuter / 0.4)
    set on_belt true
    set time_since_lane_change Min_t_btwn_lanechange
    ifelse o_ellipse = true
    [ ifelse (p_ellipse >= (random-float 1))
      [ set facial_ellipse 1 ]
      [ set facial_ellipse 0 ] ]
    [ set facial_ellipse 0 ]
  ]

  ask turtles
  [
```

```
    loop
    [ ifelse any? other turtles-here
      [ fd ((2 * (random 2)) - 1) ]
      [ stop ]
    ]
  ]

  set selected1 one-of turtles
  output-type "Selected Ped 1 = " output-print selected1
  ask selected1 [ set color orange ]
  set selected2 one-of turtles
  ask turtles [
    loop [
      ifelse (selected2 = selected1)
      [ set selected2 one-of turtles ]
      [ stop ]
  ] ]
  output-type "Selected Ped 2 = " output-print selected2
  ask selected2 [ set color yellow ]
  set selected3 one-of turtles
  ask turtles [
    loop [
      ifelse ((selected3 = selected2) or (selected3 = selected1))
      [ set selected3 one-of turtles ]
      [ stop ]
  ] ]
  output-type "Selected Ped 3 = " output-print selected3
  ask selected3 [ set color magenta ]

  set peds_in (peds_in + inc_create_stand + inc_create_walk +
inc_create_comm)
  set peds_in_belt peds_in
  set standers_in (standers_in + inc_create_stand)
  set walkers_in (walkers_in + inc_create_walk)
  set commuters_in (commuters_in + inc_create_comm)
end

;; A.11 File Output module
to setup-file
  let file user-new-file
  ;; We check to make sure we actually got a string just in case
  ;; the user hits the cancel button.
  if not is-string? file
  [ set File_Output_TimeSpace false stop ]
    ;; If the file already exists, we begin by deleting it, otherwise
    ;; new data would be appended to the old contents.
    if file-exists? file
      [ file-delete file ]
    file-open file
    ;; record the initial turtle data
  file-print "NetLogo Escalator Simulation Model Output"
  file-print ""
  file-print "INPUT PARAMETERS"
  file-print (word "Belt Speed = " BeltSpeed)
  file-print (word "Slow with Distance? = " Slow_with_Distance)
  file-print (word "Stander Pedestrian Speed = " PedSpeed_Stander)
  file-print (word "Walker Pedestrian Speed = " PedSpeed_Walker)
```

```
  file-print (word "Commuter Pedestrian Speed = " PedSpeed_Commuter)
  file-print (word "Walker Climbing Speed = " ClimbSpeed_Walker)
  file-print (word "Commuter Climbing Speed = " ClimbSpeed_Commuter)
  file-print (word "Probability of Deceleration = " Decel_Probability)
  file-print (word "Merge Space Back for Walkers = " MergeSpace_Walker)
  file-print (word "Merge Space Back for Commuters = " MergeSpace_Commuter)
  file-print (word "Lane Change Allowed? = " LaneChange_Allowed)
  file-print (word "Minimum Time Between Lane Changes = "
Min_t_btwn_lanechange)
  file-print (word "Percent Standers = " Stand_Percent)
  file-print (word "Percent Commuters out of Non-Standers = "
FastWalker_Percent)
  file-print (word "Difference in Queue Between Lanes for Arrivals to Ignore
Assignment = " Max_Queue_Diff)
  file-print (word "Ignore Initial Lane Assignment? = " Ignore_Lane_Assign)
  file-print (word "Arrivals per Hour = " Inflow)
  file-print (word "Belt Length (m) = " Belt_Length)
  file-print (word "Initially Populate Belt? = " Initially_Populate_Belt)
  file-print (word "Belt Rules? = " Belt_Rules)
  file-print (word "Percent Obeying Facial Ellipse = " Obey_Ellipse_Percent)
  file-print (word "Stop After N Seconds = " StopAfter_Seconds)
  file-print (word "Capacity Test? = " Capacity_Test)
  file-print ""
  file-print "SIMULATION RESULTS"
  file-print "Turtle Number, Breed, Ticks (seconds), xcor (tread count),
Lane, Speed (treads/tick), Desired Speed"
  file-print
"TIME,VALUE,ticks,peds_in,peds_in_belt,peds_out,Density,queue_total"
  file-print ""
end

;; A.12 GO routine
to go
  if ticks >= StopAfter_Seconds [stop]
  if (queue_left >= 90) or (queue_right >= 90) ; queue overflow
  [
    ask patches [set pcolor orange]
    stop
  ]
  lane-change
  move-belt
  arrivals
  move-queue
  tick
  departures
  create-plots
end

;; A.13 Arrivals module
to arrivals
  set inc_create_stand random-poisson (n_stand / 3600)
  create-standers inc_create_stand
  [
    set color black
    ifelse (belt_rules = "None")
    [ ifelse (queue_right - queue_left) >= Max_Queue_Diff [ set lane 1 ] [
set lane 0 ] ]
```

```
  [ set lane (random 2)]
  setxy 0 lane
  set heading 90
  set time_since_lane_change 0
  set speed_desired 0
  set speed_desired_now speed_desired
  set speed p_stand
  set on_belt false
  ifelse o_ellipse = true
  [ ifelse (p_ellipse >= (random-float 1))
   [ set facial_ellipse 1 ]
   [ set facial_ellipse 0 ] ]
  [ set facial_ellipse 0 ]
  loop
   [ ifelse any? other turtles-here
    [ fd -1 ]
    [ stop ]
   ]
 ]
 set inc_create_walk random-poisson (n_walk / 3600)
 create-walkers inc_create_walk
 [
  set color (lime + 2)
  ifelse (lanechange_allowed = false)
  [ ifelse (ignore_lane_assign = false)
   [ ifelse (belt_rules = "None")
    [ ifelse (queue_left - queue_right) >= Max_Queue_Diff [ set lane 0 ]
[ set lane 1] ]
    [ ifelse (queue_right - queue_left) >= Max_Queue_Diff [ set lane 1 ]
[ set lane 0] ]  ]
   [ set lane (random 2) ] ]
  [ ifelse ((belt_rules = "None") and (ignore_lane_assign = false))
   [ ifelse (queue_left - queue_right) >= Max_Queue_Diff [ set lane 0 ]  [
set lane 1] ]
   [ set lane (random 2) ] ]
  setxy 0 lane
  set heading 90
  set time_since_lane_change 0
  set speed_desired v_walk
  set speed_desired_now speed_desired
  set tolerable_gap_back_other (mergespace_walker / 0.4)
  set speed p_walk
  set on_belt false
  ifelse o_ellipse = true
  [ ifelse (p_ellipse >= (random-float 1))
   [ set facial_ellipse 1 ]
   [ set facial_ellipse 0 ] ]
  [ set facial_ellipse 0 ]
  loop
   [ ifelse any? other turtles-here
    [ fd -1 ]
    [ stop ]
   ]
 ]
 set inc_create_comm random-poisson (n_comm / 3600)
 create-commuters inc_create_comm
 [
```

```
    set color red
    ifelse (lanechange_allowed = false)
    [ ifelse (ignore_lane_assign = false)
      [ ifelse (queue_left - queue_right) >= Max_Queue_Diff [ set lane 0 ] [
set lane 1] ]
      [ set lane (random 2) ] ]
    [ ifelse ((belt_rules = "None") and (ignore_lane_assign = false))
      [ ifelse (queue_left - queue_right) >= Max_Queue_Diff [ set lane 0 ] [
set lane 1] ]
      [ set lane (random 2) ] ]
    setxy 0 lane
    set heading 90
    set time_since_lane_change 0
    set speed_desired v_comm
    set speed_desired_now speed_desired
    set tolerable_gap_back_other (mergespace_commuter / 0.4)
    set speed p_comm
    set on_belt false
    ifelse o_ellipse = true
    [ ifelse (p_ellipse >= (random-float 1))
      [ set facial_ellipse 1 ]
      [ set facial_ellipse 0 ] ]
    [ set facial_ellipse 0 ]
    loop
      [ ifelse any? other turtles-here
        [ fd -1 ]
        [ stop ]
      ]
  ]
  set peds_in (peds_in + inc_create_stand + inc_create_walk +
inc_create_comm)
  set standers_in (standers_in + inc_create_stand)
  set walkers_in (walkers_in + inc_create_walk)
  set commuters_in (commuters_in + inc_create_comm)
end

;; A.14 Move Queue module
to move-queue
  ;; move the queue towards the escalator
  ask-concurrent turtles with [on_belt = false]
  [
    ifelse (min-one-of other turtles with [(xcor > [xcor] of myself) and
(lane = [lane] of myself)] [distance myself]) = nobody
    [ set distance_forward 20 ]
    [ set distance_forward (distance (min-one-of other turtles with [(xcor >
[xcor] of myself) and (lane = [lane] of myself)] [distance myself]))]
    jump (min list speed distance_forward)
    loop [
      ifelse any? other turtles-here
      [ fd -1 ]
      [ stop ]
    ]
  ]

  ;; people on belt speed up and change on_belt state
  ask turtles with [ (on_belt = false) and (xcor >= 0)]
  [
```

```
    set speed (speed + v_belt)
    set on_belt true
    set peds_in_belt (peds_in_belt + 1)
    set time_since_lane_change (max list 0 (Min_t_btwn_lanechange - 2))
  ]

  ;; if we're unconcerned about the queue, we prevent queue overflow
  if Capacity_Test = true [
  set queue_removed (queue_removed + count turtles with [ xcor <= -75 ])
  ask-concurrent turtles with [ xcor <= -75 ] [die] ]

  ;; compute queue length
  set queue_left (count turtles with [(xcor < 0) and (ycor = 1)])
  set queue_right (count turtles with [(xcor < 0) and (ycor = 0)])
  set queue (queue_left + queue_right)

  set queue_total (queue + queue_removed)
  set max_queue max list queue_total max_queue
end

;; A.15 Move Belt module
to move-belt
  ask-concurrent turtles with [on_belt = true]
  [
    set rand_decel random-float 1
    if slow_with_distance = true
    [
      if (xcor >= 50)
      [ ifelse (xcor >= 75)
        [ ifelse (xcor >= 100)
          [ set speed_desired_now (max list (speed_desired - 3) 0) ]   ; slow
more after 40m run
          [ set speed_desired_now (max list (speed_desired - 2) 0) ] ] ; slow
more after 30m run
        [ set speed_desired_now (max list (speed_desired - 1) 0) ] ; slow a
bit after 20 m run
      ]
    ]

    ; FOLLOWING BEHAVIOR
    ; Determine if there is another vehicle ahead (same lane, greater x-
coordinate)
    ifelse (min-one-of other turtles with [(xcor > [xcor] of myself) and
(lane = [lane] of myself)] [distance myself]) = nobody
    ; if no vehicle, set to an aribtrarily large headway
    [ set distance_forward 20 ]
    ; if there is a vehicle, set distance_forward equal to the distance
    [ set distance_forward (distance (min-one-of other turtles with [(xcor >
[xcor] of myself) and (lane = [lane] of myself)] [distance myself]))]

    set gap_forward (distance_forward - 1)

    ; if speed >= gap, slow down to gap, keep minimum speed at v_belt
    ifelse ((speed + facial_ellipse) >= gap_forward)
    [ ifelse (rand_decel <= p_decel) ; if speed is greater than gap, we must
slow down
```

```
      [ set speed (max list (gap_forward - 1) v_belt) ] ; some probability of
extra deceleration, floor is belt speed
      [ set speed (max list (gap_forward - facial_ellipse) v_belt) ] ] ;
otherwise slow to gap, floor is belt speed
    [ ifelse (speed < (speed_desired_now + v_belt)) ; if forward gap exists,
check speed vs. speed limit
      [ ifelse (rand_decel <= p_decel)  ; if not at maximum speed
        [ set speed (max (list speed v_belt)) ] ; some probability of no
aceleration
        [ set speed (max (list (min (list (speed + 1) (gap_forward -
facial_ellipse) (speed_desired_now + v_belt) )) v_belt)) ] ] ;; accelerate in
increments of v_belt
      [ ifelse (rand_decel <= p_decel)  ; if at or above maximum speed
        [ set speed (max (list (min list (gap_forward - facial_ellipse)
(speed_desired_now + v_belt - 1)) v_belt)) ] ; some probability of
deceleration
        [ set speed (max (list (min list (gap_forward - facial_ellipse)
(speed_desired_now + v_belt)) v_belt)) ] ] ] ; otherwise maintain speed

    ;; move people at appropriate relative speed
    if belt_type_for_facial_ellipse != "Down Escalator"
    [ if facial_ellipse = 1
      [ if (any? turtles-at facial_ellipse 0)
        [ set speed (speed - facial_ellipse)  ] ] ]

    ;; move the users at appropriate relative speed
    jump speed

    ;; double check for overlap
    if any? other turtles-here
    [ fd -1 ]
  ]
end

;; A.16 Lane Change module
to lane-change
  if ( lanechange_allowed = true)
  [
    ;; COMPUTE CONSTANTS
    ask-concurrent turtles with [on_belt = true]
    [
      ;; calculate gaps around ped
      ifelse (min-one-of other turtles with [(xcor > [xcor] of myself) and
(lane = [lane] of myself)] [distance myself]) = nobody
      [ set gap_forward 20 ]
      [ set gap_forward ((distance (min-one-of other turtles with [(xcor >
[xcor] of myself) and (lane = [lane] of myself)] [distance myself])) - 1)]
      ifelse (min-one-of other turtles with [(xcor < [xcor] of myself) and
(lane = [lane] of myself)] [distance myself]) = nobody
      [ set gap_back 20 ]
      [ set gap_back ((distance (min-one-of other turtles with [(xcor <
[xcor] of myself) and (lane = [lane] of myself)] [distance myself])) - 1)]

      ;; calculate speeds of surrounding peds
      ifelse (min-one-of other turtles with [(xcor > [xcor] of myself) and
(lane = [lane] of myself)] [distance myself]) = nobody
      [ set speed_desired_now_forward 20 ]
```

```
      [ set speed_desired_now_forward ([speed_desired_now] of (min-one-of
other turtles with [(xcor > [xcor] of myself) and (lane = [lane] of myself)]
[distance myself]))]
      ifelse (min-one-of other turtles with [(xcor < [xcor] of myself) and
(lane = [lane] of myself)] [distance myself]) = nobody
      [ set speed_desired_now_back 0 ]
      [ set speed_desired_now_back ([speed_desired_now] of (min-one-of other
turtles with [(xcor < [xcor] of myself) and (lane = [lane] of myself)]
[distance myself]))]

      ;; calculate gaps in adjacent lane
      ifelse (min-one-of other turtles with [(xcor > [xcor] of myself) and
(lane != [lane] of myself)] [distance myself]) = nobody
      [ set gap_forward_other 20 ]
      [ set gap_forward_other (([xcor] of (min-one-of other turtles with
[(xcor > [xcor] of myself) and (lane != [lane] of myself)] [distance
myself])) - xcor - 1)]
      ifelse (min-one-of other turtles with [(xcor < [xcor] of myself) and
(lane != [lane] of myself)] [distance myself]) = nobody
      [ set gap_back_other 20 ]
      [ set gap_back_other (xcor - ([xcor] of (min-one-of other turtles with
[(xcor < [xcor] of myself) and (lane != [lane] of myself)] [distance
myself])) - 1)]

      ; calculate speeds in adjacent lane
      ifelse (min-one-of other turtles with [(xcor > [xcor] of myself) and
(lane != [lane] of myself)] [distance myself]) = nobody
      [ set speed_desired_now_forward_other 20 ]
      [ set speed_desired_now_forward_other ([speed_desired_now] of (min-one-
of other turtles with [(xcor > [xcor] of myself) and (lane != [lane] of
myself)] [distance myself]))]
      ifelse (min-one-of other turtles with [(xcor < [xcor] of myself) and
(lane != [lane] of myself)] [distance myself]) = nobody
      [ set speed_desired_now_back_other 0 ]
      [ set speed_desired_now_back_other ([speed_desired_now] of (min-one-of
other turtles with [(xcor < [xcor] of myself) and (lane != [lane] of myself)]
[distance myself]))]
    ]

    ;; LANE CHANGE RIGHT
    ask-concurrent turtles with [(on_belt = true) and (ycor = 1) and
(time_since_lane_change >= Min_t_btwn_lanechange)] [
      ifelse ((speed_desired_now < speed_desired_now_back) and (gap_back <=
(speed_desired_now_back - speed_desired_now)))
      [
        ;; lane change while tailgated
        if ((not any? turtles-at 0 -1)) ;; see if space exists
        [
          set lane 0
          set facial_ellipse 0
        ]
      ]
      [
        ;; standard lane change
        if ((speed_desired_now >= gap_forward) and (gap_forward_other >
gap_forward) and (gap_back_other >= tolerable_gap_back_other)
```

```
        and ((speed - v_belt) <= gap_forward_other) and (gap_back_other >=
tolerable_gap_back_other)) [

        if (not any? turtles-at 0 -1) ;; see if space exists
        [ set lane 0 ] ]
      ]
    ]

    ;; LANE CHANGE LEFT  ;; test if in right lane, time is over minimum;
also, standers not allowed to merge left
    ask-concurrent turtles with [(on_belt = true) and (ycor = 0) and
(time_since_lane_change >= Min_t_btwn_lanechange) and (breed != standers)]
    [
      if ((speed_desired_now >= gap_forward) and (gap_forward_other >
gap_forward) and (gap_back_other >= tolerable_gap_back_other)
        and ((speed - v_belt) <= gap_forward_other) and (gap_back_other >=
tolerable_gap_back_other))
      [

        if (not any? turtles-at 0 1)
        [ set lane 1 ]
      ]
    ]

    ;; PERFORM LANE CHANGES SIMULTANEOUSLY
    ask-concurrent turtles with [(on_belt = true) and (lane != ycor)]
    [
      set ycor lane
      set time_since_lane_change 0
    ]

    ;; increase time since lane change
    ask turtles with [on_belt = true]
    [ set time_since_lane_change (time_since_lane_change + 1) ]

  ]
end

;; A.17 Departures module
to departures
  set peds_out (peds_out + count turtles with [xcor >= treads])
  set walkers_out (walkers_out + count walkers with [xcor >= treads])
  set standers_out (standers_out + count standers with [xcor >= treads])
  set commuters_out (commuters_out + count commuters with [xcor >= treads])
  if ([xcor] of selected1) >= treads
  [
    set selected1 min-one-of turtles [xcor]
    ask turtles [
      loop [
        ifelse ((selected2 = selected1) or (selected1 = selected3))
        [ set selected1 one-of turtles ]
        [ stop ]
    ] ]
    output-type "Selected Ped 1 = " output-print selected1
    ask selected1 [ set color orange ]
  ]
  if ([xcor] of selected2) >= treads
```

```
  [
    set selected2 min-one-of turtles [xcor]
    ask turtles [
      loop [
        ifelse ((selected2 = selected1) or (selected2 = selected3))
        [ set selected2 one-of turtles ]
        [ stop ]
    ] ]
    output-type "Selected Ped 2 = " output-print selected2
    ask selected2 [ set color yellow ]
  ]
  if ([xcor] of selected3) >= treads
  [
    set selected3 min-one-of turtles [xcor]
    ask turtles [
      loop [
        ifelse ((selected3 = selected1) or (selected2 = selected3))
        [ set selected3 one-of turtles ]
        [ stop ]
    ] ]
    output-type "Selected Ped 3 = " output-print selected3
    ask selected3 [ set color magenta ]
  ]
  ask-concurrent turtles [ if xcor >= treads [die] ]
  set flow (peds_out / ticks * 3600)
  if v_belt > 0 [
  ifelse ticks >= (treads * 5 / v_belt)
  [ set flow_max max list flow flow_max ]
  [ set flow_max "stabilizing..." ] ]
  set density ((peds_in_belt - peds_out) / Belt_Length)
end

;; A.18 Create Plots module
to create-plots
  ;; compute system average speeds
  set avg_speed_stand ((sum [speed] of standers) / (standers_in -
standers_out) * .4)
  set avg_speed_walk ((sum [speed] of walkers) / (walkers_in - walkers_out) *
.4)
  set avg_speed_comm ((sum [speed] of commuters) / (commuters_in -
commuters_out) * .4)

  ;; plot speeds
  set-current-plot "Speeds"
  set-current-plot-pen "avg Stand"
  plot avg_speed_stand
  set-current-plot-pen "avg Walk"
  plot avg_speed_walk
  set-current-plot-pen "avg Comm"
  plot avg_speed_comm

  ;; plot fundamental characteristics
  set-current-plot "Fundamental Characteristics"
  set-current-plot-pen "Walkers in System"
  plot (walkers_in - walkers_out)
  set-current-plot-pen "Standers in System"
  plot (standers_in - standers_out)
```

```
  set-current-plot-pen "Commuters in System"
  plot (commuters_in - commuters_out)
  set-current-plot-pen "Density (ped/10 m)"
  plot (density * 10)
  set-current-plot-pen "Queue Length"
  plot queue_total
  set-current-plot-pen "Flow (1000 ped/hr)"
  plot (flow / 1000)

  ;; plot time-space diagram
  set-current-plot "Time-Space"
  set-current-plot-pen "Ped1"
  plot (([xcor] of selected1) * 0.4)
  set-current-plot-pen "Ped2"
  plot (([xcor] of selected2) * 0.4)
  set-current-plot-pen "Ped3"
  plot (([xcor] of selected3) * 0.4)

  ;; display time passed
  set time ticks

  ;; output file parameters
  if file_output_timespace = true [
  file-print (word "TIME,VALUE," ticks "," peds_in "," peds_in_belt ","
peds_out "," Density "," queue_total)
  foreach sort turtles [
  ask ? [
    file-print (word who "," breed "," ticks "," xcor "," Lane "," Speed  ","
speed_desired_now)
  ] ] ]
end
```

Appendix B – Model Operation Instructions

The instructions contained within this section are taken directly from the support documentation included with the moving belt system model. All documentation related to this thesis is currently hosted online at http://dropbox.peterpages.net/thesis, with the final version of the model hosted at http://dropbox.peterpages.net/thesis/Version_1.1.2.html. The information contained within this section can be used to operate that model.

B.1 VERSION 1.1

This model was created between November 17-19, 2010 by Peter Kauffmann. It is based on Version 1.0.3, released November 16, 2010. For internal record-keeping purposes, the specific version is 1.1.2.

Improvements from Version 1.0.3 to 1.1.2 include:

- the ability to plot time-space diagrams for selected pedestrians
- the option to create a text file containing model parameters, system information, and pedestrian characteristics at every tick for use in subsequent data analysis. This data can be used to make more detailed plots as well. This option does not function when operated within a web browser, only through the NetLogo software.

B.2 OPERATION

1) To configure the model, use the various inputs to set the user characteristics and composition, operational parameters, and belt characteristics. Alternatively, select a default scenario to automatically set these inputs to a realistic set of values. Descriptions of the example scenarios are given below.
2) Press "Setup" to generate the simulation environment, including creating a belt at the user-specified length and populating it with users.
3) Press "Go" to begin the simulation. To slow down the model, move the slider above the simulation window to the left to decrease the number of model time steps ("ticks") that occur per actual second.
4) For an even better view of the tick-by-tick operation of the model, press "Go" again to stop the simulation and instead advance it one tick at a time by pressing the "Go Once" button.

More information about the model's operations can be found further down the page.

B.3 EXPLANATION

After the parameters are set by the user in the upper section, the model is generated using the "Setup" routine. A belt is generated at the user-specified length with alternating colors shown every 0.4 meters, representing the code-specified tread depth on an escalator in the United States according to ASME A17.1. The input parameters are also used to populate the belt so that the flow is roughly accurate from the start.

After pressing "Go," the model generates escalator riders assuming Poisson distributed arrivals at the user-specified arrival rate and composition. These riders are placed on the model using a simplified linear queuing assumption at the base of the moving belt. These users move at the specified walking speed until reaching the belt, where they are assigned a "speed limit" equal to the sum of the belt speed (v_belt = 0 for stairs) plus their on-belt speed, which can be specified based on whether the belt is an escalator or moving walk as well as local parameters.

While on the belt, the user speed is determined by the available space in front of them in conjunction with the maximum speed that the individual is able to travel as limited by the previously calculated "speed limit.” Passing behavior is loosely based on TRANSIMS automotive passing rules with allowances made for pedestrian-specific behaviors and aggressiveness defined by the user. At the end of the belt, riders are tabulated and removed from the simulation environment.

B.4 EXAMPLE SCENARIOS

Various example scenarios have been created to show the implementation of escalator management strategies and different hypothesized aspects of pedestrian behavior. The code name for each scenario is composed of several different components. For example:

```
 T N V  D  2500  0.4
 | | |  |  |     |     |
 | | |  |  |     |     Other parameters noted here
 | | |  |  |     Belt Speed (m/s)
 | | |  |  Initially simulated arrival rate (passengers/hour)
 | | |  Belt Type (Down escalator/Up escalator (with ellipse observer %)
                  /Moving walk)
 | | Ignore Initial Lane Assignment? (V=check/X)
 | Belt Rules (None/Walk only/Stand only)
 Lane Change Allowed? (True/False)
```

The calculated approximate capacity values were determined by turning on the "capacity test" option. This prevents queue overflow from halting the model. Simulations were then run at various arrival rates until the maximum steady-state condition was reached.

==== Scenario 0 - TNV D 2500 0.4 ====
Basic escalator operation: downhill operation, passing allowed, no belt rules, and randomized walker/commuter lane assignment. Volume is kept low so passing behavior can be observed. Slowing with distance is ignored.
Approximate Capacity: 6100 ped/hr, which closely matches the 5800 ped/hr value from the 80% rule.
Lane changing will be turned off for subsequent scenarios in order to reduce variability in the results.

`==== Scenario 1 - FNX D 3500 0.4 ====`
Original operation of this model before implementation of passing behavior: downhill, no passing, no belt rules, strict lane assignment. Observe how the system's capacity is entirely based on that of the individual lanes and the average speed of walkers and commuters by varying the walking percentage and the climbing speeds.
-- Approximate Capacity: 7100 ped/hr. Notice how by removing passing behavior we make the system more streamlined.

`==== Scenario 2 - FNX U-95 3500 0.4 ====`

Same as Scenario 1 except on an uphill escalator with 95% of passengers observing the facial ellipse (0.4m forward gap required for comfort). Notice how the riders space themselves out and the resulting capacity drop.
-- Approximate Capacity: 3700 ped/hr, since the belt space is not being utilized as effectively. Also, note the lowered density.

`==== Scenario 3 - FNX U-95 3500 0.8 ====`
Same as Scenario 2 except with higher belt speed of 0.8 m/s versus 0.4 m/s. This speed is used in the Moscow subway because of their extremely deep stations. See how capacity is improved - and actually doubled - by a faster belt even in a case where capacity has been lowered by the required pedestrian spacing.
-- Approximate Capacity: 7500 ped/hr, assuming a simplified boarding delay.

`==== Scenario 4 - FSX U-95 3500 0.4 ====`
Same as Scenario 2 except with a rule implemented where users may only stand on the belt - no walking/climbing is allowed. Observe the effect on capacity.
-- Approximate Capacity: 4400 ped/hr, which is slightly higher because of the removal of imperfections in packing due to walking.

`==== Scenario 5 - FWX U-95 3500 0.4 ====`
Same as Scenario 2 (and Scenario 4) except with a rule implemented where all users must walk up the escalator. Although this would be impractical to enforce in the real world, observe the effect on packing efficiency and capacity.
-- Approximate Capacity: 3800 ped/hr - still slightly higher than Scenario 2 because of the reduction of variability due to the presence of standers, but lower than packing in with standers alone.

`==== Scenario 6 - FNX U-0 3500 0.4 SLOW ====`
Same as Scenario 1 except that it accounts for the endurance of the pedestrians. Based on speed curves found in the literature, pedestrians will slow as they climb an escalator. Note that this must therefore take place on an upwards escalator, but the percentage of pedestrians observing the facial ellipse has been set to zero to remove the impact of this factor. Because of the discrete nature of this CA model, speeds must be reduced in increments of 0.4 m/s (1 cell/second), and the first of these drops takes place at 20m, which is ten treads from the exit of the escalator.
-- Approximate Capacity: 5100 ped/hr. This reduction is caused by a shockwave present at the speed drop, which eventually propagates back causing a capacity drop relative to Scenario 1.

B.5 ERROR CODES

Several error codes exist to display problems with a model run.

- If a feature is selected that is not yet supported by the model, the background will turn red.
- If there is a queue overflow beyond what is supported at the base of the belt (currently 100 people), the background will turn solid orange.
- If an unsupported button is pressed, the background will turn blue.

B.6 FUTURE IMPROVEMENTS

If time permits, I plan to construct and implement a more realistic representation of the bottleneck that users experience while boarding the escalator. My thoughts on this protocol would be to use NetLogo's potential for continuous rather than discrete simulation to generate users in a Poisson-distributed manner at a random location some distance away from the mouth of the bottleneck and set them at a trajectory and speed that will take them towards the belt entrance. More complicated pedestrian spacing rules would need to be developed to govern the entrance to the bottleneck, but hopefully this would more accurately model the entry behavior seen in Hoogendoorn's "Pedestrian Behavior at Bottlenecks" (2005).

The starting setup would look something like this, with pedestrians being generated on the double lines and queuing into the belt:

```
===========
||
||           __
||             ====
||           __====
||
||
===========
```

More documentation on this improvement will be included at its release or in the proposed research section, depending on my committee's decision.

B.7 RELATED MODELS

General inspiration was taken from the "Traffic 2 Lane" model in the NetLogo library, although it must be said that there are several bugs with that model, most notably that vehicles will not change lanes even though the model description says they should. http://ccl.northwestern.edu/netlogo/models/Traffic2Lanes

The only code within my model that can be sourced directly to "Traffic 2 Lane" are the method by which the POPULATE-BELT procedure adjusts pedestrians that are placed on an already occupied cell and the way that the average speed of the various entities are calculated, although I have had to modify the math involved in each of these processes to fit the configuration of my model. Additionally, the method by which the user can select pedestrians for the time-space diagram is influenced by the SELECTED-CAR routine of that model.

The general code structure by which the data is written to an external file was inspired by the "File Output Example" tutorial model, available in NetLogo's internal Model Library.

B.8 REVISION HISTORY

This model was created between November 17-19, 2010 by Peter Kauffmann. It is based on Version 1.0.3, released November 16, 2010. For internal record-keeping purposes, the specific version is 1.1.2.

Improvements from Version 1.0.3 to 1.1.2 include:

- the ability to plot time-space diagrams for selected pedestrians
- The option to create a text file containing model parameters, system information, and pedestrian characteristics at every tick for use in subsequent data analysis. This data can be used to make more detailed plots as well. This option does not function when operated within a web browser, only through the NetLogo software.

Changes from Version 0.2.4 to 1.0.3 include:

- several default scenarios for analysis of rule-based and physical system constraints (described below)
- inclusion of a "capacity test" option
- support for pedestrian passing behavior with user defined aggressiveness
- improved rules for when pedestrians ignore their initial lane assignment

The proposed bottleneck simulation has been suspended from this simulation and will be attempted in a side project if time permits for future implementation in this model.

Modifications from Version 0.2.2 to 0.2.4 include:

- ability for users to ignore lane assignment if queue difference is too high
- accounting for the surprisingly important difference between "distance to forward pedestrian" and "gap to forward pedestrian"
- more realistic following rules for close interaction
- acceleration behavior based on the literature instead of my faulty "infinite acceleration" assumption
- generally, this model has a better implementation of following than v.0.2.2, which can be seen by the fact that it is capable of passing a realistic number of passengers for a given belt speed

Changes from Version 0.1 to 0.2.2 included:

- the inclusion of default scenarios (only one so far)
- the implementation of pedestrian following behavior. Before, pedestrians would just walk right through one another, which is obviously inaccurate but took some time to implement into the model
- support for rules that constrain passengers to walk-only or stand-only behavior
- the option to have climbers "tire" as they ascend stairs
- the ability to specify the minimum forward free space and the percentage of passengers who obey this rule in order to model the "facial ellipse" that causes riders to increase the forward space on a downwards traveling escalator

Revisions from Version 0.0 to 0.1 included:

- the option to initially populate the belt with users
- improved output displays and plots.
- math errors (dividing by zero, mostly) involved in setting the belt speed to 0 to model stairs have been addressed

B.9 CREDITS AND REFERENCES

Updates to this model should be available at http://dropbox.peterpages.net/thesis/