Technical Report: An Agent Based Simulation of Autonomous Pods

Robert Tinn

1 Summary

In recent years, technological advances have led to the development of autonomous transportation systems which have the potential to be safer and more efficient than current solutions, by reducing dependencies on expensive and unreliable humans. These pods operate in pedestrianised zones and represent a 'last mile' transport solution for commuters. This report explains and summarises the design of the agent control algorithms.

2 Added Pedestrian Behaviour

The pedestrian agents had additional behaviour added so that they can interact realistically with the pod agents. The new attributes allow pedestrians to:

- Determine if they are on a collision course with a pod
- Calculate the time to when that collision will occur
- Take evasive action if the collision will occur soon (or they are nearly touching the pod)
- Navigate around the pod keeping a specified distance away from it

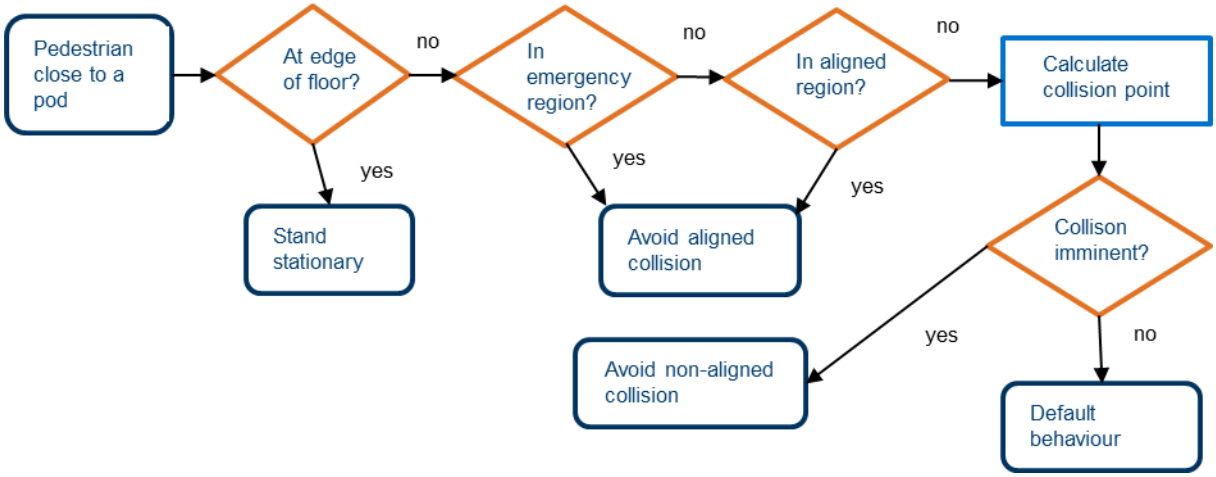

Figure 1: Pedestrian control algorithm

2.3 Pod Agent Modelling

For modelling the autonomous pod, a unique agent is created which does not use any of MassMotion's default agent behaviour. Figure 2 illustrates the attributes and methods that are

defined by the pod class. A pod is given a very small, finite minimum speed so it always has a valid heading (heading is equivalent to a normalised velocity vector). The pods minimum circle is implemented by the ‘constrain heading’ method which prevents large deviations in direction. Otherwise the other main attributes and methods should be self-explanatory.

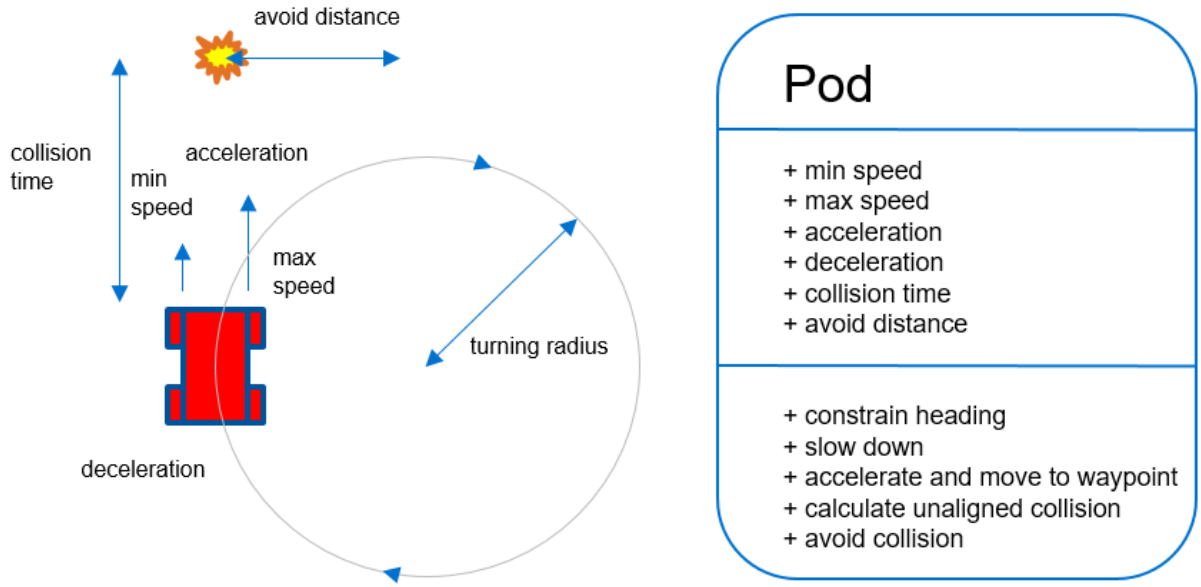

Figure 2: Pod class illustration

Figure 3: Collision detection regions for close pod and pedestrian agents

Two collision detection algorithms are employed for pod and pedestrian agents. Figure 3 demonstrates the two different regions, agents must be located in, to activate the different collision functions. ‘Non-aligned collision detection’ calculates the intersection of the two agent’s headings, and the projected position for the pedestrian when the pod has reached the intersection, so the pod can adjust its heading to avoid a collision. ‘Aligned collision detection’ is required because the default pedestrian agents in MassMotion do not travel in straight lines, they wander in a natural fashion, which makes it difficult to calculate collisions in this instance. Pedestrians have two ‘aligned’ regions, as opposed to just one for a pod, to allow them to move aside for a pod to pass, if one is following behind them.

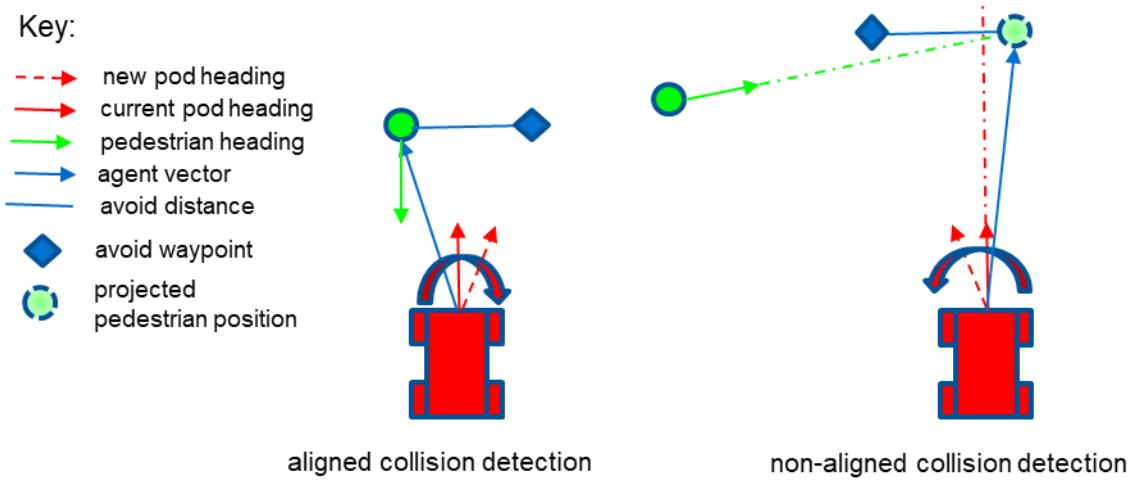

Figure 4: Sketch of collision detection functions