Chapter 4

Intersection Control for Automated Vehicles

4.1 Introduction

The capacity and safety of an intersection can be considered in terms of the number and type of conflict points which it contains. When every entrance to the intersection is connected to every exit by a smooth arc, a conflict point exists wherever two arcs intersect. They can be classified as head-on, following, crossing, diverging or merging. In many studies these points are assumed to be fixed, and only speed of the vehicles which controls the rate of progress along the arcs is varied to minimise delay.

With the development of path planning around obstacles on-the-fly, it becomes possible to consider what arrangement of conflict points is best given the traffic at a particular instant in the near future.

4.2 Simulation

Platooning with speed choice by a centralized controller was implemented with a vehicle to intersection messaging scheme. The full site is divided into zones, each one containing a single intersection. Each AGV in the fleet has a copy of the roadmap which is static. The fleet controller interfaces with the warehouse management system to get the next material transfer job, consisting of a pick location and a drop location. All jobs are assumed to be of unit size and each AGV has a capacity of one unit. With these assumptions, a straightforward policy is to assign the next job to the AGV nearest to the pick location - first-come-first-served scheduling. When an AGV receives a new job, it finds the shortest path through the roadmap using the Floyd-Warshall algorithm [36]. Next it must send its planned path to the intersection controller for the zone it currently occupies. The intersection controller stores the plan and current position of every AGV approaching the conflict point of the intersection. Every time it receives a new plan it must recalculate the approach speed for every approaching AGV to minimize total travel time without collision. This will happen every time an AGV enters the zone from somewhere else, or an AGV within the zone is assigned a new job.

The intersection controller was implemented based on [37]. The surrounding lanes are first discretized into segments. The intersection shown in Figure 4.1 is divided into six segments, each of length 10 meters. The critical segments are the two that cross in the center. There are two routes defined, one starting on the left and traveling to the right and the other starting at the bottom and traveling up. One AGV takes route 1

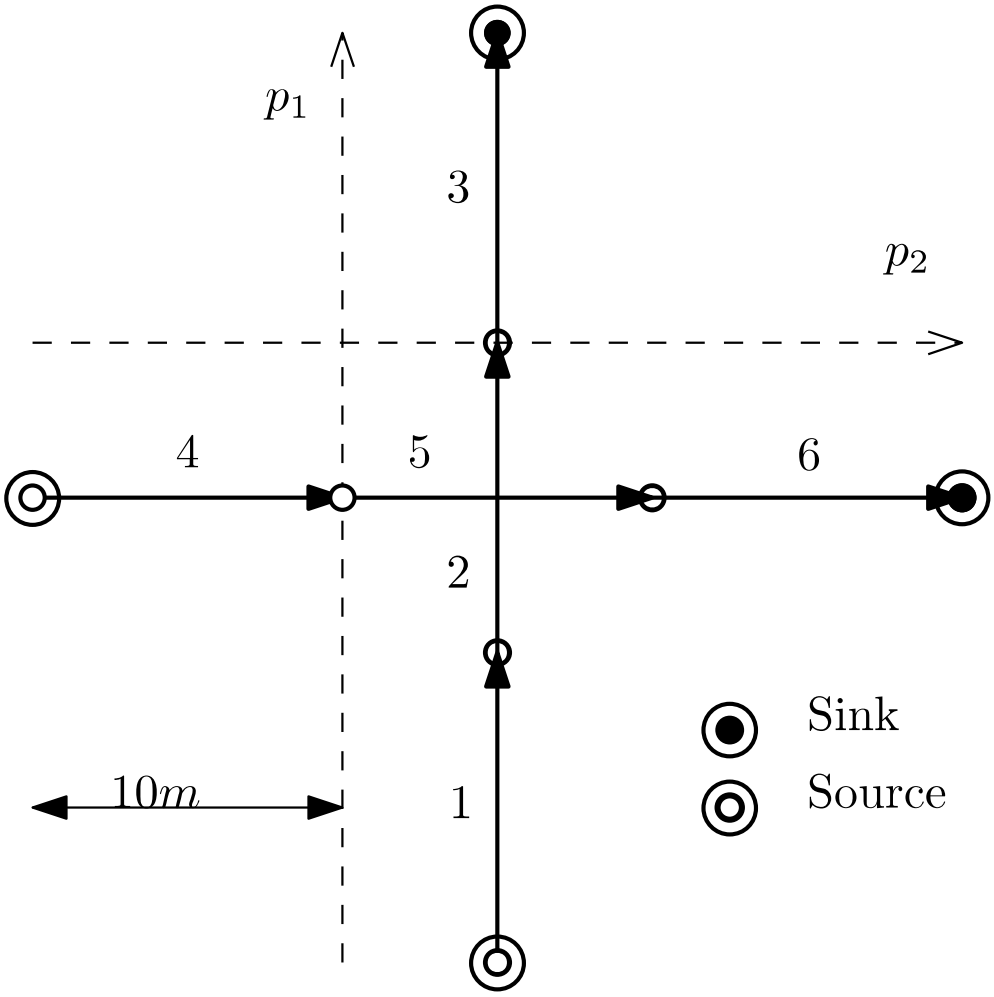

Figure 4.1: Intersection layout with two conflicting routes.

and the other takes route 2. If they both travel at maximum speed they will collide in the center.

The dynamic model for each AGV assumes they are able to exactly follow the path, and attempt to reach the target speed for each segment subject to a limited rate of acceleration of am/s^2.

The ApproachPlan message sent by the AGV contains a sequence of segments which it intends to traverse, along with its current distance along the first one. The flow of messages is shown in Figure 4.2. The SpeedList sent by the intersection controller contains the optimal speed for every segment in the plan. The speeds can be found with the nonlinear program in Equation 4.9.

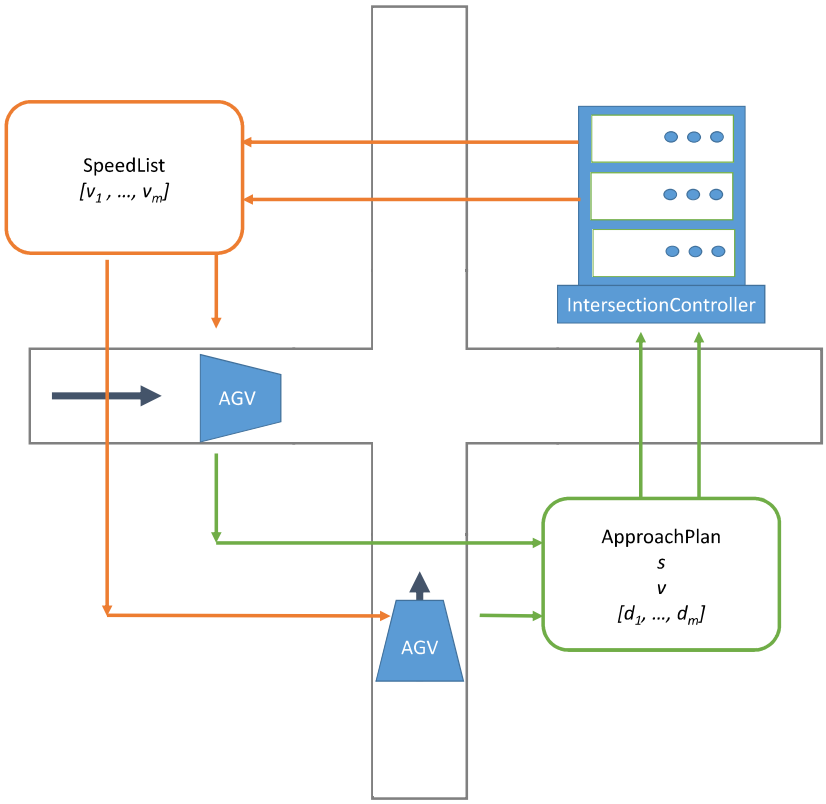

Figure 4.2: Messages exchanged by participants approaching intersection.

4.2.1 Motor Dynamic and Electrical Model

The simulated AGV are based on a small payload 100kg total mass, with a maximum speed of 10m/s and peak acceleration of 5m/s^2, allowing them to stop safely within 10 metres. The intersection controllers use a constant acceleration model to calculate the time and space deadlines they pass to the AGV. To make the simulation test worthwhile a simulation model with slightly more complexity was used to evaluate the performance.

The brushless DC electric motor used in [38] has suitable properties to propel a 100kg battery powered vehicle. A DC motor can be modelled by the steady state equivalent circuit shown in Figure 4.20 which is well known, for example see [39].

This simple circuit leads to the relationship between current and internal resistance given by Equation 4.1.

$$V_{cc} = E_D + I_a R_a \tag{4.1}$$

It should be noted that the brushless DC motor is sometimes grouped with AC machines

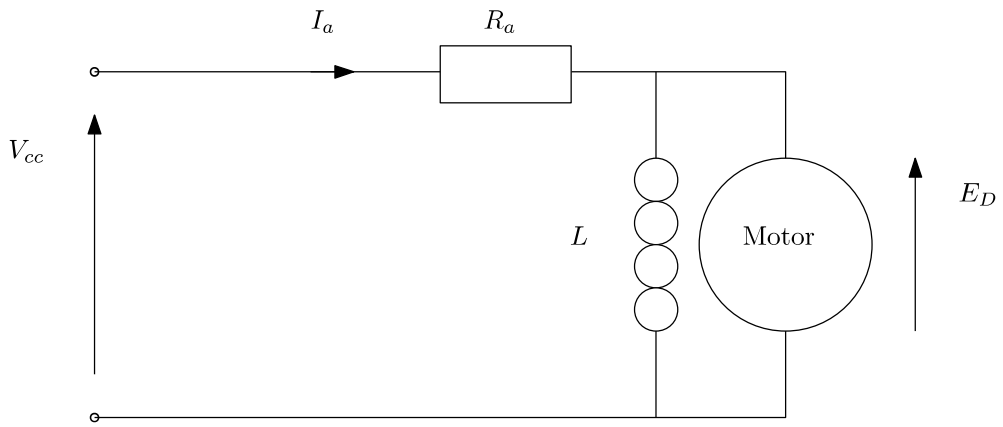

Figure 4.3: Steady state equivalent circuit for a DC motor.

such as the induction motor due to its varying excitation [40]. The coils are energised in order to keep the magnetic field at 90 degrees to the field generated by the permanent magnets mounted on the rotor.

The field strength of the magnets, the number of poles and the number turns of the armature coils can be captured in the motor constant k_T relating torque to armature current.

$$\tau = k_T I_a \tag{4.2}$$

$$E_D = \frac{rpm}{k_e} \tag{4.3}$$

There are numerous loss sources in an electric motor such as winding resistance, flux leakage, eddy currents in the core and so on [40]. By using real-world measured mechanical power output and electrical input, an equivalent winding resistance R_a for the simple model can be found. The parameters are shown in Table 4.6.

4.2.2 Air Resistance

In addition to the resistive losses in the motor windings losses due to air resistance were also modelled. AGVs typically operate at low speed ¡3m/s, hence the unaerodynamic

Table 4.1: Motor parameters used in simulation.

k_v [rpm/v]	6
k_T [Nm/A]	1.53
P_{mech}@375rpm [kW]	3.6
P_{elec}@375rpm [kW]	6.37
$^*R_a$ [Ohms]	1.0

shape of many models. However, the target speed of 10m/s is quite high around 30 miles per hour and drag could become significant.

The drag coefficient C_{drag}=1 for a cuboid shape was used, taken from [41]. The frontal area A of $1m^2$ is consistent with a box shape for maximum load carrying capacity. The density of air at room temperature is close to $1kg/m^3$.

$$F_D = C_{drag}\rho v^2 A \tag{4.4}$$

4.2.3 Arrival Distribution

Previous work in road traffic modelling has used a variety of point distributions to model arrivals. A good summary is given in [42] or the chapter on Microsimulation in [43].

Learning from Road Transport Models

One option is to assume arrivals at a point are completely independent of each other, but occur at an average rate for the time of day which has been measured by inductive loop placed in the road. These assumptions lead to a Poisson distribution such as that shown in Figure 4.4. This can be generated computationally by drawing a sequence of numbers from a uniform distribution and applying Equation **??** to compute the time delta until the next arrival. To validate the arrivals drawn in this way, we check the log plot of the frequency against time delta is linear and the median is equal to the specified rate as shown in Figure 4.7.

The assumption that arrivals are independent does not hold if traffic density is high. This is because vehicles slow down to keep a safe distance from the one in front of them. The simplest modification to the poisson distribution to account for this is to discard time deltas which violate the specified safe headway, leading to the shifted exponential

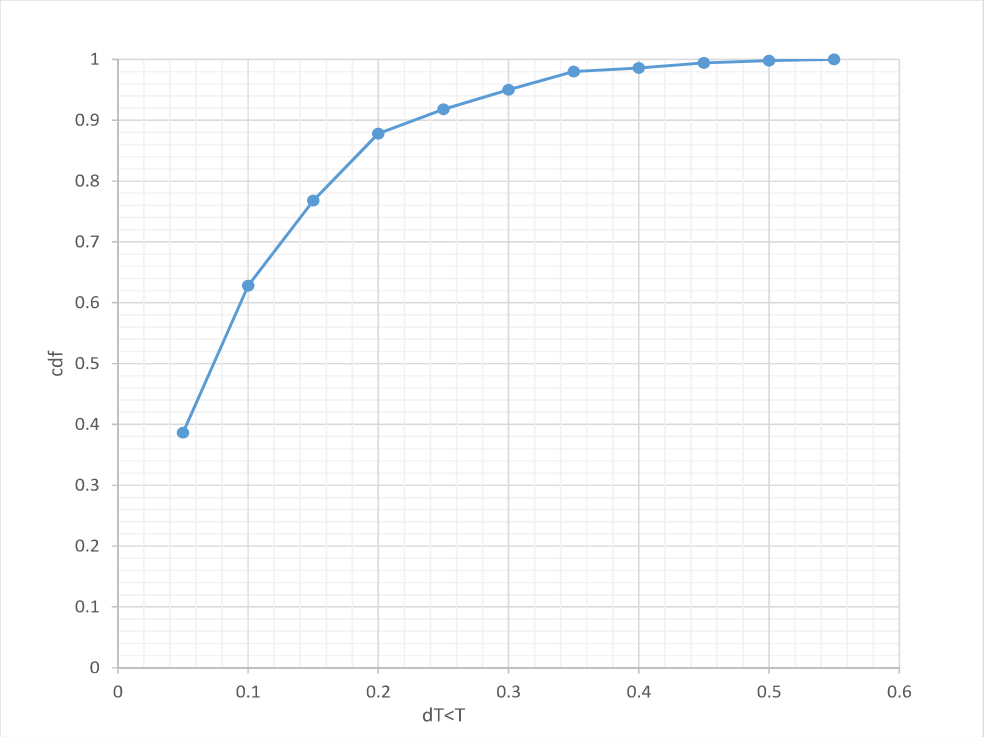

Figure 4.4: Cumulative Distribution Function of arrivals following a Poisson distribution.

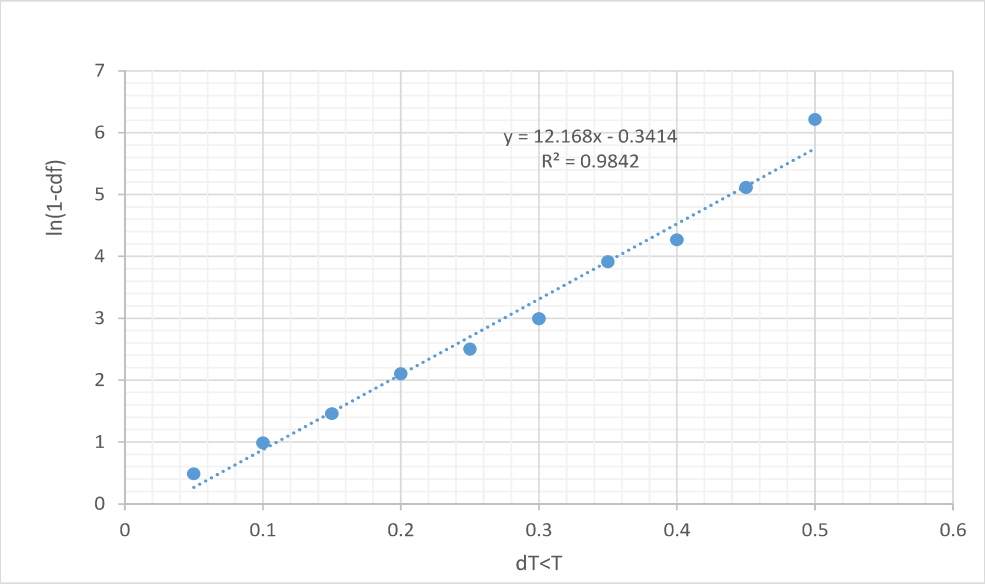

Figure 4.5: Log Cumulative Distribution Function of arrivals with linear fit.

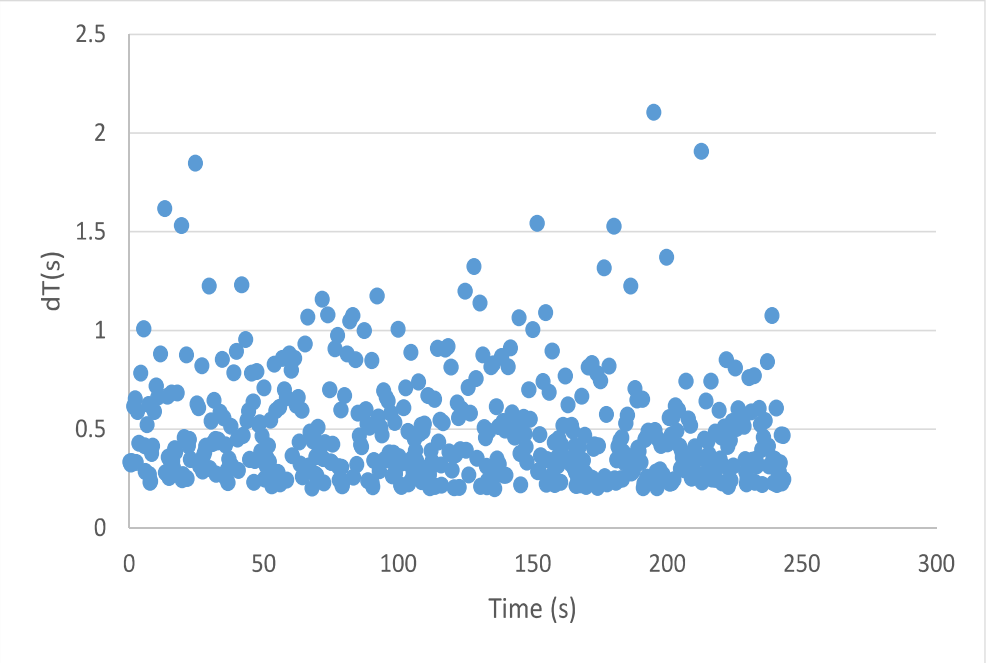

Figure 4.6: Scatter plot of 500 arrival time deltas drawn from a shifted exponential distribution with a minimum headway of 0.2 seconds.

distribution. A scatter plot of times from this distribution is shown in Figure 4.6. Now the arrivals will always be realistic in the sense vehicle will not overlap/arrive unsafely but the variance will be smaller. The two parameters the average arrival rate λ and the minimum safe headway τ together control the variance. This is because the median of the distribution must be λ, so if $1/\tau$ is close to λ the variance must be very small.

Unique Features of AGV Traffic

The arrival distribution corresponds to the properties of AGV traffic satisfying the following assumptions:

- More than one AGV is permitted to enter the same link if they are travelling in the same direction.
- Entry to links in the roadmap is denied if the number already present on the link reaches a fixed capacity

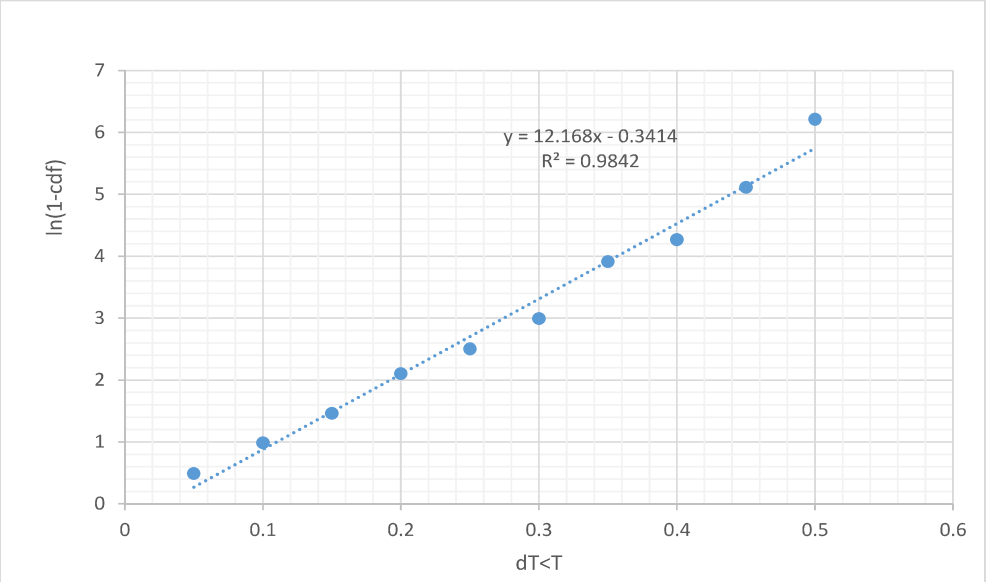

Figure 4.7: Log Cumulative Distribution Function of arrivals with linear fit.

- Car-following behaviour within the link is governed by the frontal safety system on board each AGV. This provides an outer (warning) zone which, if tripped, the AGV will slow to wm/s at a_wm/s and an inner zone (emergency)), which if tripped the vehicle will come to a complete stop at a_em/s, where $|a_e|$ ¿ $|a_w|$
- The capacity of a link is the number of AGVs which would fit without the inner front safety zone of each being tripped by the one in front. This can be calculated for n AGVs each with safety zone length d_e as $n_c = n\dot{d}_w$
- Every AGV is informed whether it is able to proceed before it reaches the end of its current link. The AGV will keep slowing down to stop at the end, until it is informed there is space on the next link, at which point it will accelerate to full speed.

According to these assumptions, a suitable model is the shifted exponential, with a minimum headway based on the size of the AGV safety zone.

Arrival Variation Based on Approach Lane Occupancy

Another consideration is how the traffic on the approach lanes should feed back into the arrival times. Based on the counting semaphore for each link assumption, this can

be modelled with two queues at the entryway. The lane queue and the busy queue. If the occupancy of the lane is less the n_c, new arrivals are added to the lane queue at full speed, in the simulation interval exceeding their point arrival time, at a position they would have reached by the simulation time. If the occupancy of the lane is more than n_c new arrivals are added to the back of the busy queue. At the next time step where $n < n_c$, the AGV at the front of the busy queue is moved onto the lane. The arrival speed is reduced according to acceleration rate a_W based on the time between the arrival and the time step where is can proceed. For longer time periods the new arrival will have zero speed. All arrivals which were not at the front of the busy queue will arrive at the warning speed v_w=0.3m/s.

4.2.4 Division of Responsibility Between Intersection and AGV Controllers

With different signal-free intersection control approaches, the functionality carried out centrally by the fixed-server compared to that by the mobile AGV varies significantly. Of particular interest are the interfaces used by Digani et al [37] where the intersection provides average speeds for discrete segments, and the one used by Levin and Rey [29] where the reservation message sent by the intersection contains the arrival time at the conflict point.

The arrival time at the conflict point is similar to a widely used waypoint interface. As the time is the important quantity for ensuring collisions are avoided this interface seems more sensible. However, on its own it is not sufficient to ensure FIFO behaviour of AGVS approaching in the same lane. Many studies assume an external mechanism for car-following based on local sensors such as the Intelligent Driver Model in [44].

Without a car-following or platooning approach locally on the AGV, simple methods to manage cross traffic can easily fail to maintain FIFO ordering. This problem is often encountered in traffic simulation with cellular models and there are various techniques to avoid it without generating a trajectory for each vehicle [45].

[29] and [37] do not mention any local car following behaviour and seek to avoid all collisions by the centralised application of constraints. This makes it difficult to compare them to a semaphore based intersection controller, unless car following is used in combination with semaphore access control.

4.3 Methods For Zone-Based Intersection

4.3.1 Intersection Controller Objective

The objective is to minimize J_T the total travel time for all vehicles. It is convenient for exposition to optimize over the inverse of speed of each segment $\phi_k = 1/v_k$. Vehicle i submits a plan containing m_i segments before the conflict and n_i segments in conflict. The control model is based on the average speed of each approaching AGV over each segment. This is to simplify the description of the intersection controller, and assist analysis. More sophisticated motion models could take the place of Equation 4.10 and Equation 4.12 to create a similar type of problem with a convex travel time objective and non-convex constraints. The parameters for one vehicle can be collected in the vector $\boldsymbol{\phi}_i$ as shown in Equation 4.5

$$\boldsymbol{\phi}_i = [\phi_1, ..., \phi_{(m_i+n_i)}]^T \tag{4.5}$$

The parameters for p vehicles each traversing $(m_i + n_i)$ segments are assembled into a vector as in Equation 4.6

$$\boldsymbol{\phi} = [\boldsymbol{\phi}_\mathbf{1}^T, ..., \boldsymbol{\phi}_p^T]^T \tag{4.6}$$

Similarly, the length of each segment in plan i can be arranged into a vector

$$\boldsymbol{d}_i = [d_1, ..., d_{(m_i+n_i)}]^T \tag{4.7}$$

and collected for p vehicles into a vector as in Equation 4.8.

$$\boldsymbol{d} = [\boldsymbol{d}_\mathbf{1}^T, ..., \boldsymbol{d}_p^T]^T \tag{4.8}$$

This leads to the minimum travel time objective in Equation 4.9.

$$\begin{gathered} \min_{\boldsymbol{\phi}} \boldsymbol{J_T} = \boldsymbol{d}^T\boldsymbol{\phi} \\ \text{subject to} \\ \boldsymbol{\phi}_{max} > \boldsymbol{\phi} > \boldsymbol{\phi}_{min} \\ \boldsymbol{\phi}^T \boldsymbol{H}_{i,j} \boldsymbol{\phi} > \mathbf{0} \quad \forall i, j \in [1, p] \quad \text{with } j > i \end{gathered} \tag{4.9}$$

The condition $j > i$ in Equation 4.9 indicates that the number of constraints varies with the number of vehicles p as $\frac{p(p-1)}{2}$. This corresponds to one constraint between each pair of approaching AGVs.

4.3.2 Intersection Controller Timing Constraints

By definition, each intersection has a single conflict zone, the union of all segments which intersect there. This makes it possible to express the constraint that vehicles do not collide in terms of time. Vehicle i arrives at the first conflicted segment ω_i^{min} and departs from the last at ω_i^{max}. The following three subsections set out three alternative ways of expressing the collision avoidance constraints which have been evaluated. The arrival time is given by Equation 4.10. Considering average speeds, the departure time ω_i^{max} is also linear, this is given by Equation 4.12.

$$\omega_i^{min} = \sum_{k=1}^{m_i} \boldsymbol{d}_i[k]\boldsymbol{\phi}_i[k] = \boldsymbol{e}^T\boldsymbol{\phi}_i \tag{4.10}$$

where

$$\boldsymbol{e}[k] = \begin{cases} \boldsymbol{d_i}[k] & \forall k < m_i \\ 0 & \text{otherwise} \end{cases} \tag{4.11}$$

and m_i is the number of segments on the path of vehicle i before arrival at the conflicted segment.

$$\omega_i^{max} = \omega_i^{min} + \sum_{i=1}^{n_i} \boldsymbol{d}_i[k]\boldsymbol{\phi}_i[k] = \boldsymbol{f}^T\boldsymbol{\phi}_i \tag{4.12}$$

where

$$\boldsymbol{f}[k] = \begin{cases} \boldsymbol{d_i}[k] & \forall k < m_i + n_i \\ 0 & \text{otherwise} \end{cases} \tag{4.13}$$

and n_i is the number of segments on the path of vehicle i which are conflicted. Note that Equations 4.10 and 4.12 only depend on the $\boldsymbol{\phi}_i$ of vehicle i.

Linear FIFO Constraints

If the order in which the AGV cross the conflict zone is fixed to be First-In-First-Out, the timing constraint is linear. For a pair where the leader enters the conflict at t_i and takes Δt_i seconds to cross at maximum speed the condition for entry time of the follower t_{i+1} given by Equation 4.14.

$$-t_i + t_{i+1} \leq \Delta t_i \tag{4.14}$$

There is one constraint between each adjacent pair so $p-1$ constraints total for p vehicles. These can be expressed in the form $\boldsymbol{A}_{ub}\phi \leq \boldsymbol{b}_{ub}$ as in Equation 4.15. This is

correct for two AGV arranged in distance order, each traversing one approach and one conflict segment.

$$\begin{bmatrix} -d_1 & d_2 & \mathbf{0} \\ \vdots & \ddots & \vdots \\ \mathbf{0} & -d_{p-1} & d_p \end{bmatrix} \begin{bmatrix} \phi_1 \\ \vdots \\ \phi_p \end{bmatrix} \leq \begin{bmatrix} \Delta t_1 \\ \vdots \\ \Delta t_{p-1} \end{bmatrix} \tag{4.15}$$

The timing constraint between each pair of vehicles can be expressed with a modulus operator as in Equation4.16.

$$|\alpha_i - \alpha_j| > \beta_i + \beta_j \tag{4.16}$$

Here

$$\alpha_i = \omega_i^{max} + \omega_i^{min} \tag{4.17}$$

represents the midpoint of the time vehicle i occupies the conflicted segment and

$$\beta_i = \omega_i^{max} - \omega_i^{min} \tag{4.18}$$

represents the range of the time either side of the midpoint, both scaled by a factor of two.

In matrix form

$$\alpha_i = \boldsymbol{f}^T \boldsymbol{\phi}_i + \boldsymbol{e}^T \boldsymbol{\phi}_i = \mathbf{1}_i^T \boldsymbol{A} \phi_i \tag{4.19}$$

with $\boldsymbol{A} = diag(\boldsymbol{f} + \boldsymbol{e})$

$$\beta_i = \boldsymbol{f}^T \boldsymbol{\phi}_i - \boldsymbol{e}^T \boldsymbol{\phi}_i = \mathbf{1}_i^T \boldsymbol{B} \phi_i \tag{4.20}$$

with $\boldsymbol{B} = diag(\boldsymbol{f} - \boldsymbol{e})$

The resulting linear program (with parameters $\in \mathbb{R}$) has $p - 1$ constraints as each AGV is only constrained by the preceding one.

Quadratic Constraints

Another way to treat the modulus operator in Equation 4.16, without forcing any particular arrival order is to square both sides as to give the expression in Equation 4.21.

$$\alpha_i^2 - \alpha_j^2 - 2\alpha_i\alpha_j - (\beta_i^2 + \beta_j^2 + 2\beta_i\beta_j) > 0 \tag{4.21}$$

Collecting terms by subscript gives

$$(\alpha_i^2 - \beta_i^2) - (\alpha_j^2 + \beta_j^2) - 2(\alpha_i\alpha_j + \beta_i\beta_j) > 0 \tag{4.22}$$

The matrix $\mathbf{\Lambda}_i j$ captures the constraints between a pair of vehicles and always contains four sub-matrices as shown in in Equation 4.54. It is compatible with $\boldsymbol{\phi}_{i,j} = [\boldsymbol{\phi}_i^T, \boldsymbol{\phi}_j^T]^T$, containing only the relevant speeds for vehicles i and j.

$$\mathbf{\Lambda}_{ij} = \begin{bmatrix} \mathbf{\Lambda}_{ij}^{ii} & \mathbf{\Lambda}_{ij}^{ij} \\ \mathbf{\Lambda}_{ij}^{ji} & \mathbf{\Lambda}_{ij}^{jj} \end{bmatrix} \tag{4.23}$$

Expanding

$$\begin{aligned} \begin{bmatrix} \boldsymbol{\phi}_i^T, \boldsymbol{\phi}_j^T \end{bmatrix} \begin{bmatrix} \mathbf{\Lambda}_{ij}^{ii} & \mathbf{\Lambda}_{ij}^{ij} \\ \mathbf{\Lambda}_{ij}^{ji} & \mathbf{\Lambda}_{ij}^{jj} \end{bmatrix} \begin{bmatrix} \boldsymbol{\phi}_i \\ \boldsymbol{\phi}_j \end{bmatrix} \\ = \boldsymbol{\phi}_i^T \mathbf{\Lambda}_{ij}^{ii} \boldsymbol{\phi}_i + \boldsymbol{\phi}_j^T \mathbf{\Lambda}_{ij}^{jj} \boldsymbol{\phi}_j + \boldsymbol{\phi}_i^T \mathbf{\Lambda}_{ij}^{ij} \boldsymbol{\phi}_j + \boldsymbol{\phi}_j^T \mathbf{\Lambda}_{ij}^{ji} \boldsymbol{\phi}_i \end{aligned} \tag{4.24}$$

makes it possible to compare terms with the scalar expression in Equation 4.22. This leads to the following expressions for the submatrices in Λ in terms of $\alpha_i = \mathbf{1}_T \boldsymbol{A}_i \boldsymbol{\phi}_i$ and $\beta_i = \mathbf{1}_T \boldsymbol{B}_i \boldsymbol{\phi}_i$

$$\mathbf{\Lambda}_{ij}^{ii} = (\boldsymbol{A}_i - \boldsymbol{B}_i) \mathbf{1}_i \mathbf{1}_i^T (\boldsymbol{A}_i - \boldsymbol{B}_i) \tag{4.25}$$

$$\mathbf{\Lambda}_{ij}^{jj} = -(\boldsymbol{A}_j + \boldsymbol{B}_j) \mathbf{1}_j \mathbf{1}_j^T (\boldsymbol{A}_J + \boldsymbol{B}_j) \tag{4.26}$$

$$\mathbf{\Lambda}_{ij}^{ij} + \mathbf{\Lambda}_{ij}^{ijT} = -2(\boldsymbol{A}_j + \boldsymbol{B}_j) \mathbf{1}_j \mathbf{1}_i^T (\boldsymbol{A}_i + \boldsymbol{B}_i) \tag{4.27}$$

For more than two vehicles this can be arranged into a block diagonal matrix $\boldsymbol{H}_{ij}$ which is compatible with the input parameters, but still only represents the constraints between a pair.

Expressed in this way it is clear the constraints are quadratic and it is trivial to differentiate twice to find the Hessian is the stack of constraint matrices $[\boldsymbol{H}_{ij}, \ldots]$. The objective is certainly convex as it is linear but the constraints may not be. If the Hessian of the constraints is positive semi-definite then they are convex and interior point methods will either find the global optimum or prove that there is no feasible solution [46]. The Hessian depends on the parameters of the roadmap, the number of approaching vehicles and their distance from the conflict.

Mixed Integer Constraints

A third way of treating the timing constraint in Equation 4.16, also without forcing any particular arrival order involves splitting each constraint into two based on the sign of $(\alpha_i - \alpha_j)$ as shown in equation 4.28. Again, this is expressed in terms of the midpoint

α_i, α_j and extent β_i,β_j of the time when vehicle i and vehicle j occupy the segment on their own path which passes through the conflict point,

$$\begin{aligned} \alpha_i - \alpha_j &> \beta_i + \beta_j \quad \text{if } \alpha_i > \alpha_j \\ \alpha_i - \alpha_j &< -(\beta_i + \beta_j) \quad \text{otherwise} \\ \text{where } &\alpha_i, \alpha_j, \beta_i, \beta_j > 0 \end{aligned} \tag{4.28}$$

In order to apply these OR constraints, an additional integer parameter b_k can be introduced for each pair of AGV, along with an arbitrary large number M as shown in Equation 4.29.

$$\begin{aligned} \alpha_i - \alpha_j + b_{i,j}M &> \beta_i + \beta_j \\ \alpha_i - \alpha_j - (1 - b_{i,j})M &< -(\beta_i + \beta_j) \\ \text{where } b_{i,j} &\in [0, 1] \\ M >> \alpha_i, \alpha_j, \beta_i, \beta_j & \end{aligned} \tag{4.29}$$

Now the problem is combinatorial rather than convex and appropriate methods must be used. These may be based on exhaustive search such as Branch-and-Bound, or solving a sequence of convex relaxations of the original problem [47]. Combinatorial problems quickly become intractable for large numbers of variables, but in this case the underlying problem of arrival order is combinatorial, so exhaustive methods are needed to find the global minimum.

It is possible to further assume every AGV travels at maximum speed once it reaches the conflict, simplifying equation Equations 4.17 and 4.18 with a constant $p_i = g_i\phi_{min}$.

$$\alpha_i = 2\omega_i^{min} + p_i \tag{4.30}$$

$$\beta_i = p_i \tag{4.31}$$

where g_i is the length of the conflicted segments in the plan of vehicle i. In this case Equation 4.29 can be restated

$$\begin{aligned} \omega_i^{min} - \omega_j^{min} + b_{i,j}M &> p_j \\ \omega_i^{min} - \omega_j^{min} - (1 - b_{i,j})M &< -p_i \\ \text{where } b_{i,j} &\in [0, 1] \end{aligned} \tag{4.32}$$

4.4 Methods For Conflict Point Intersection

For a intersection between roads with multiple lanes, it makes sense to plot the centreline of each lane, and the arc of each turning motion between lanes. Any point where two

of these arcs cross is called a conflict point. By controlling arrival time at a conflict point, collisions can be avoided, while vehicles that do not pass the same conflict point can both proceed [29].

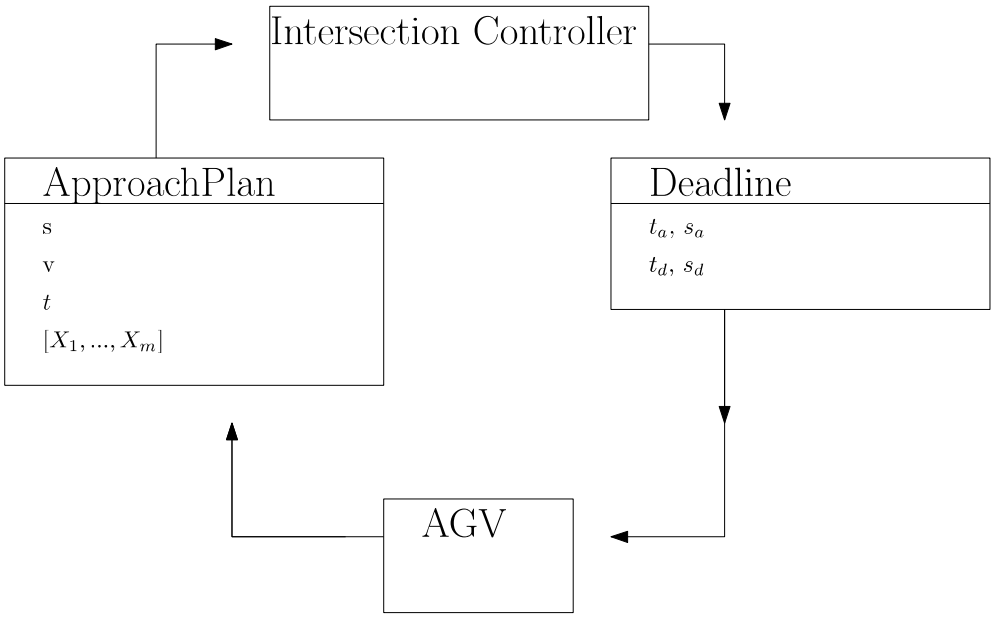

Figure 4.8: Key messages of the "Plan/Deadline" interface governing access to a conflict point intersection.

A controller based on a common heuristic for controlling access to a shared resource, the binary semaphore, is included for comparison with the optimal methods. Similar approaches are widespread in AGV control [48] and [49].

4.4.1 Semaphore Approach

To implement the "plan/deadline" interface, a binary semaphore for each conflict point was needed. The semaphore controller does not compute the entry and exit times of the conflict as it has is no model of the vehicle dynamics. It simply raises the semaphore, sending a "full speed ahead" message to the closest AGV to the intersection and a position deadline to all others.

For this reason the deadline message contains the position at which the AGV must stop, without any timing. The AGV must stop at the given position until it receives a 'proceed' message form the intersection controller.

Table 4.2: Parameters bounds used to generate test scenarios.

Parameter	High Bound	Low Bound
T_C [s]	0.5	0.1
$\frac{1}{\lambda} = \Delta T_a$ [s]	10	2

	λ_1	λ_2	f
HLHT	0.5	0.5	2
HLMT	0.1	0.5	2
HLLT	0.1	0.1	2
LLHT	0.5	0.5	10
LLMT	0.1	0.5	10
LLLT	0.1	0.1	10

Table 4.3: Parameters for test scenarios. All units s^-1. Each scenario is identified with the first two characters relating to the latency between periodic messages from the intersection controller where High Latency is 500ms and Low Latency is 100ms and the second two relating to the arrival rate, where High Traffic has λ=10 arrivals per second on both approaches, Low Traffic has λ=2 arrivals per second on both, and Mixed Traffic has one lane with λ_1=10 and the other with λ_2=2. For example High Latency, High Traffic becomes HLHT.

4.5 Performance Comparison

Both conflict point representation and the zone representation are identical if the intersection has exactly one conflict. On this basis the semaphore approach and two versions of minimum crossing time optimal control, one with FIFO linear constraints and the other with quadratic constraints were compared on a simulated intersection comprising two lanes which cross in the middle. The approach distance is fixed at 15 metres.

The controllers will be tested with traffic levels and control latencies intended to bound the likely values of these parameters, shown in Table 4.2.

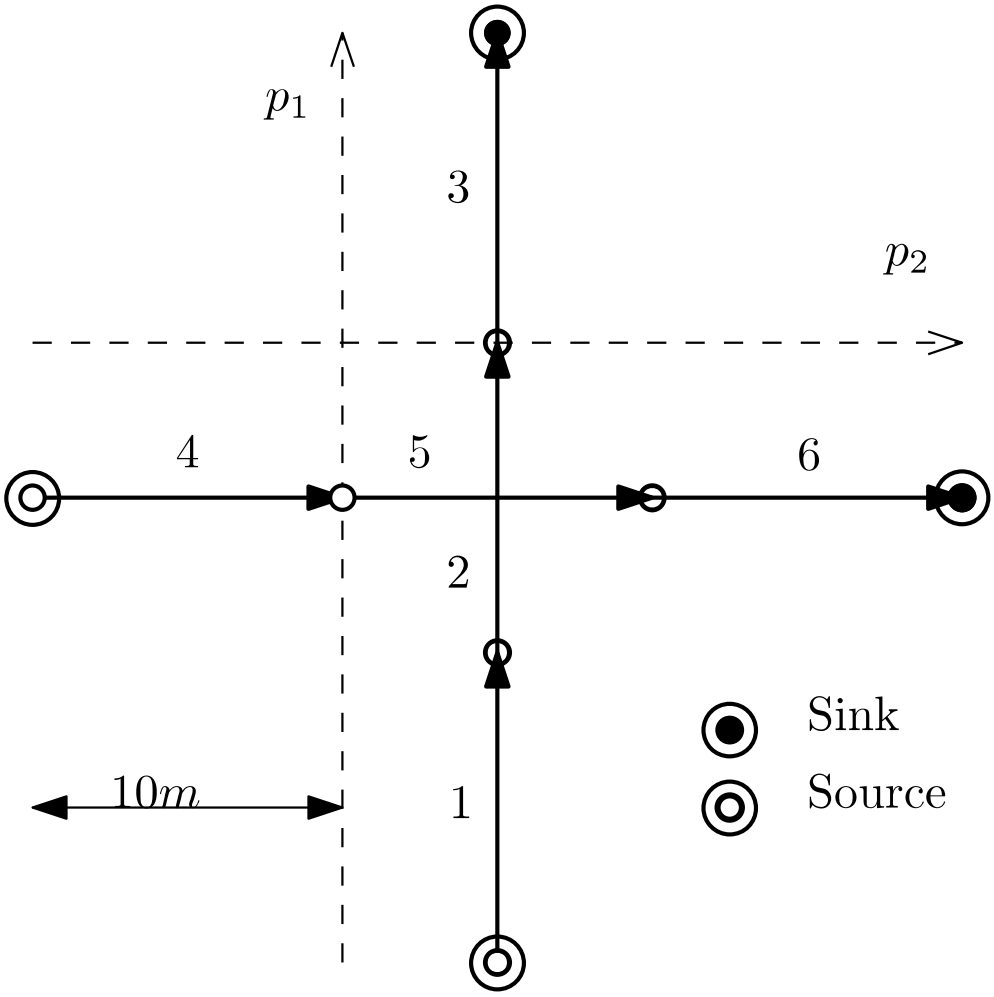

Figure 4.9: Intersection layout with two conflicting routes.

4.6 Simulation Results

The different approaches to intersection control were evaluated on a simulation of a simple intersection, comprised of two 30m lanes which cross in the middle as shown in Figure 4.21. There are two entrances to the map, one at the start of each lane. By varying the arrival rate λ and the update frequency f, six scenarios were created with the parameters shown in Table 4.11.

	λ_1	λ_2	f
HLHT	0.5	0.5	2
HLMT	0.1	0.5	2
HLLT	0.1	0.1	2
LLHT	0.5	0.5	10
LLMT	0.1	0.5	10
LLLT	0.1	0.1	10

Table 4.4: Parameters for test scenarios. All units s^-1. Each scenario is identified with the first two characters relating to the latency between periodic messages from the intersection controller where High Latency is 500ms and Low Latency is 100ms and the second two relating to the arrival rate, where High Traffic has λ=10 arrivals per second on both approaches, Low Traffic has λ=2 arrivals per second on both, and Mixed Traffic has one lane with λ_1=10 and the other with λ_2=2. For example High Latency, High Traffic becomes HLHT

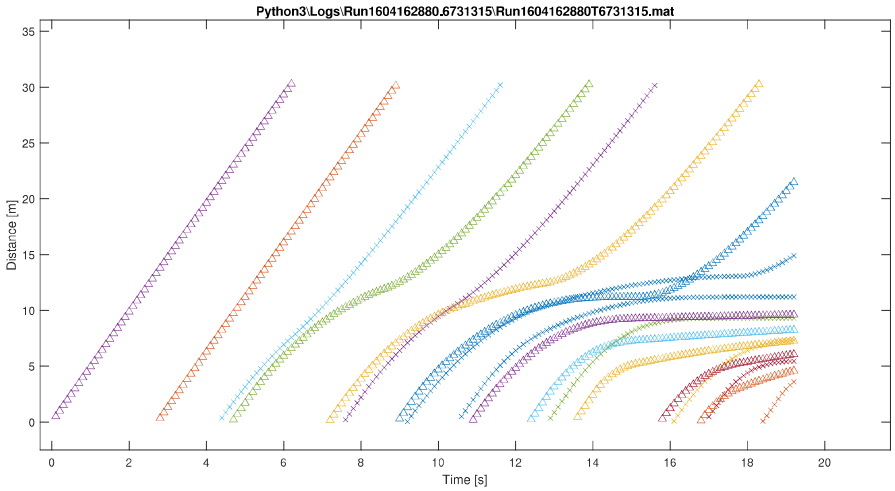

Figure 4.10: Position time series for the semaphore heuristic controller in scenario 1.

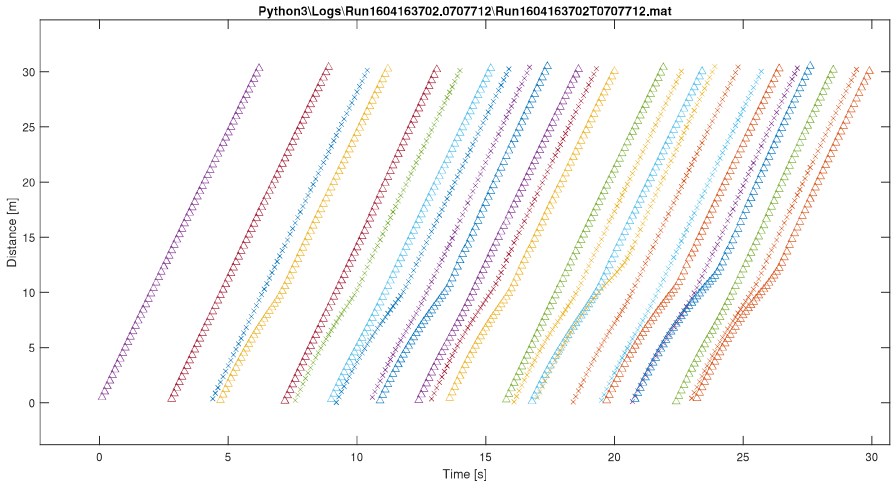

Figure 4.11: The position time series for the FIFO Optimal Controller in scenario 1.

4.6.1 Trajectory Comparison with Fixed Arrival Pattern, 30 Second Run

The distance-time plot for the semaphore controller is shown in Figure 4.10. The distance axis is measured from the start of the submitted plan for the AGV. All plans start 15 metres from the conflict point on both lanes. AGV travelling in the x-direction are shown with 'x' markers, while those travelling in the y-direction are shown with triangle markers. The simulation did not proceed beyond 20 seconds because of a collision with a new arrival.

The distance-time plot for the FIFO Optinal Controller is shown in Figure 4.11. It was able to proceed for the full 30 seconds without any collision as the approach lane did not fill up. This only shows that the throughput of the FIFO optimal control is higher than semaphore method, and is sufficient to meet the demand of $\lambda = 0.5$ on each lane, without a queue forming.

In these two runs the arrival sources were not linked to the occupancy af the approach lane, so if the approach lane fills up, and average speed drops, so the new arrivals can appear right on top of vehicle at the back of the queue, leading to a collision which the interseciton controller is unable to prevent. This is not a realistic crash situation because in a real site approaching vehicles will detect the back of queue with their front safety sensor and stop in time.

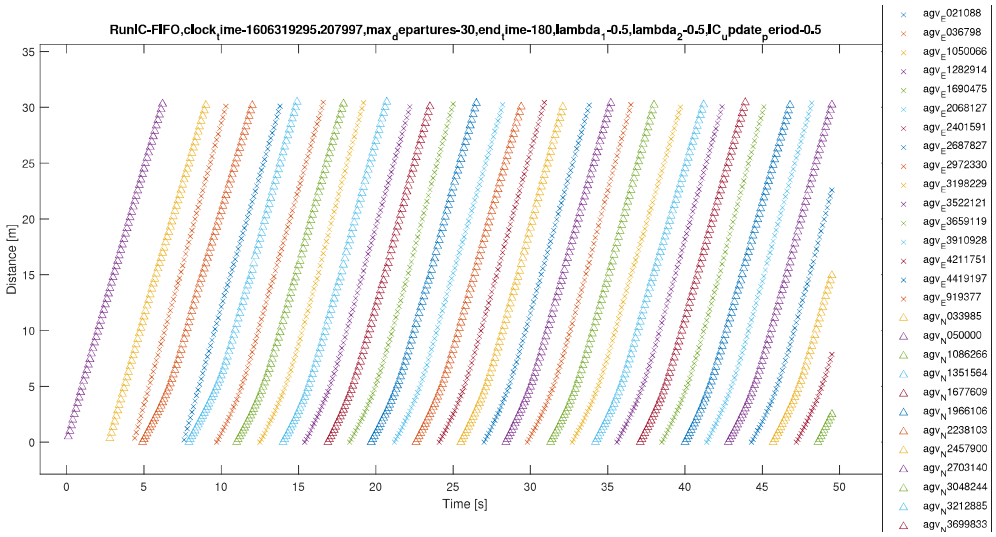

Figure 4.12: The position time series for the FIFO Optimal Controller in the High Traffic High Latency Scenario

In Figure 4.11 it can be seen that 24 vehicles pass through the intersection in 30 seconds. This is very close to the upper limit of one vehicle per second determined by on the arrival rate.

4.6.2 Trajectory Comparison with Dynmaic Arrival Pattern, 30 Departures Run

To understand more about the performance of the controllers the arrival pattern was linked to the appraoch lane occpancy. Rather than deal with the complexity of speed reductions due to safety sensors, the model was based on a roadmap reservation system which are widely usedin centralized or decentralized form. The capacity of the approach lane (10 metres long) was set to 1 AGV. If the static arrival pattern suggested an arrival while the lane was full, it entered an arrival queue. At the next simulation time step where the approach lane had capacity, if there was any vehicles in the queue, the first would be introduced to the simulation at the start of the approach lane with zero speed (as opposed to the random arrivals, which enter the lane at full speed).

With arrival lane capacity of one AGV, 30 AGVs were able to cross the intersection in 82.3 seconds with the semaphore control. With FIFO optimal control this was reduced to 49.6 seconds. This is a reduction of 39.7%, entirely due to centralized speed

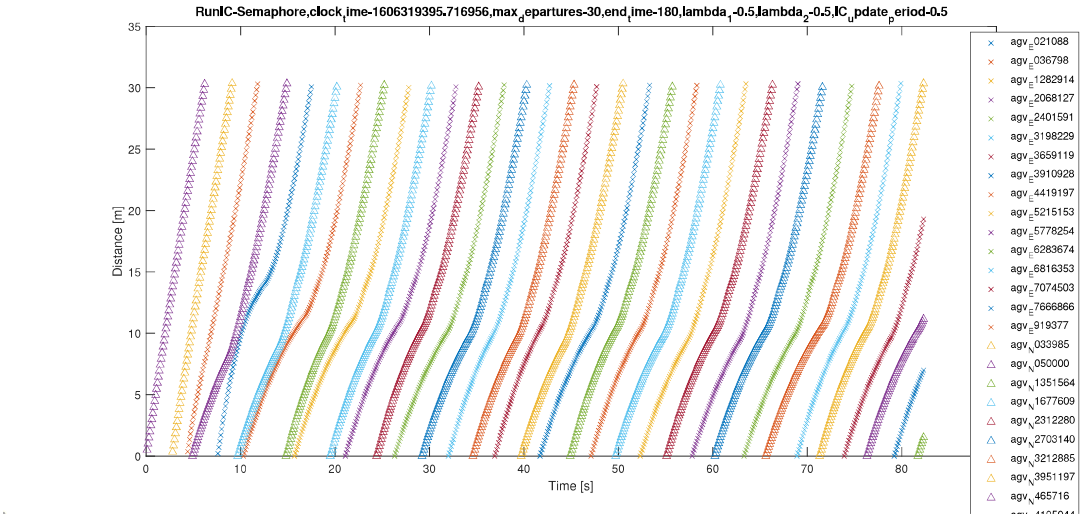

Figure 4.13: The position time series for the Semaphore Controller in the High Traffic High Latency Scenario.

choice, as both used FIFO ordering. The centralized optimal FIFO controller is able to predict when the conflict will become unoccupied, so following vehicles don't need to slow down as much as with Semaphore control.

The free flow time to cross 30 metres at maximum speed of 5 metres per second is six seconds. The mean travel time across the Semaphore controlled intersection was 309.1, indicating a delay of 309.1/30 - 6 = 4.30 seconds. The delay due to the FIFO Controlled intersection was 195.1/30 - 6 = 0.50s. This dramatic reduction might be expected given the objective for the FIFO optimal method was to minimise total travel time.

4.6.3 Energy Consumption

The use of an approach speeds minimising total travel time increases the energy consumption per vehicle as shown in Table 4.6.3. The extra information about the departure time of leading vehicles should lead to reduced energy expense slowing down. As total energy usage (and therefore usage per vehicle) actually increase, and air resistance was not considered, it suggests the higher average speeds reduce the efficiency of the motor.

	Semaphore	FIFO Optimal
Total Electrical Energy (MJ)	542.310	641.170
Total Mechanical Energy (MJ)	61.549	68.448
Completion Time (s)	82.3	49.6
Total Travel Time (s)	309.1	195.1

Table 4.5: Energy and Completion Time Results

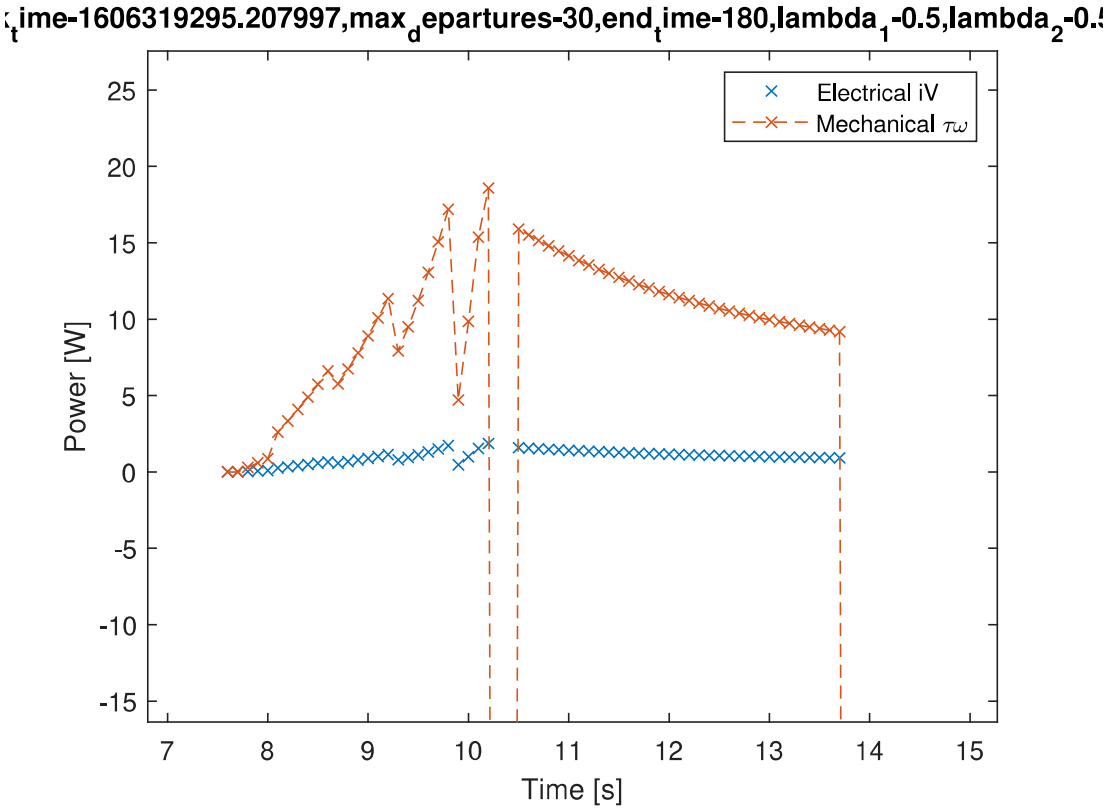

Figure 4.14: The power dissipated over time for one AGV under FIFO control.

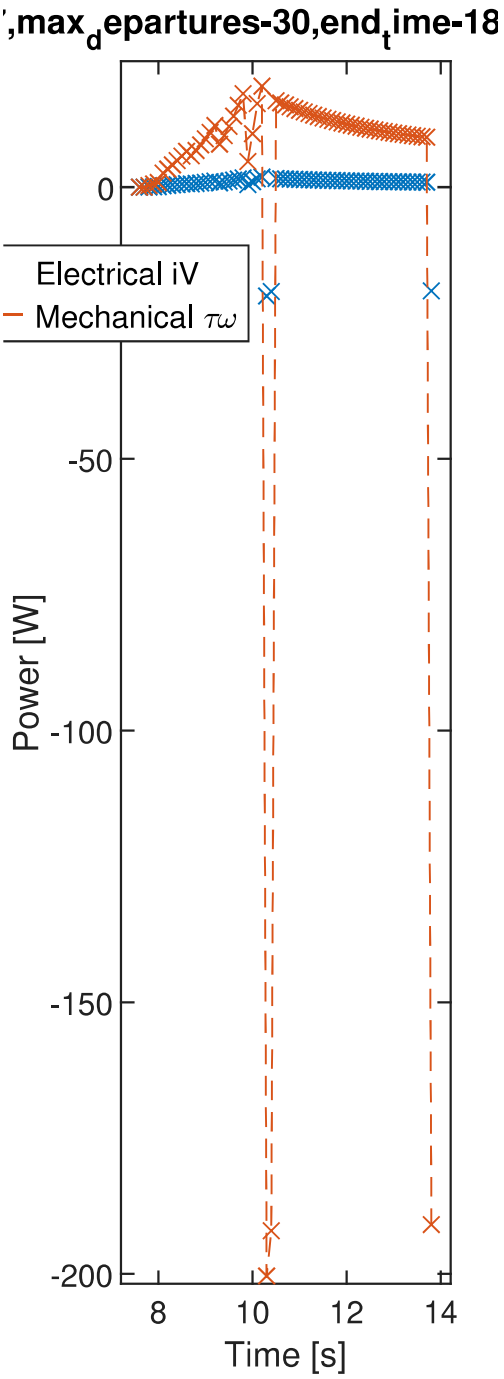

Figure 4.15: Spurious data points of the power dissipated over time for one AGV under FIFO control.

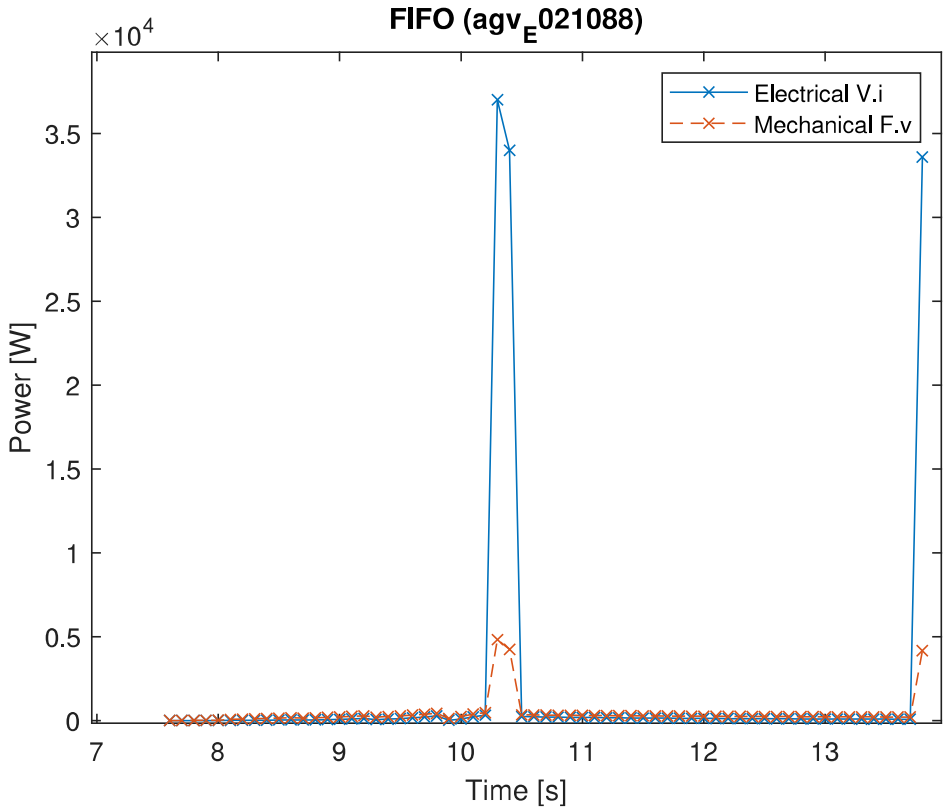

Figure 4.16: The electric power was recalculated from the logged value of armature current i^2R with a resistance of R=0.001ω. The mechanical power weas recaluclated using the abs(motive_force*velocity).

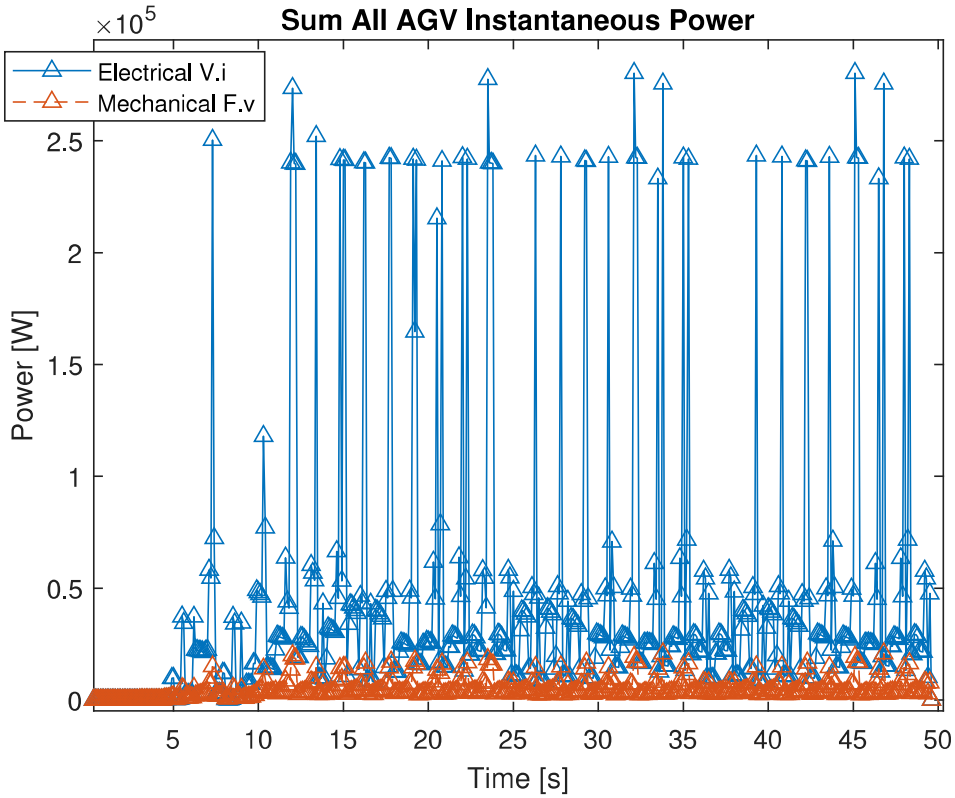

Figure 4.17: The power dissipated over time for all AGVs in total under FIFO control.

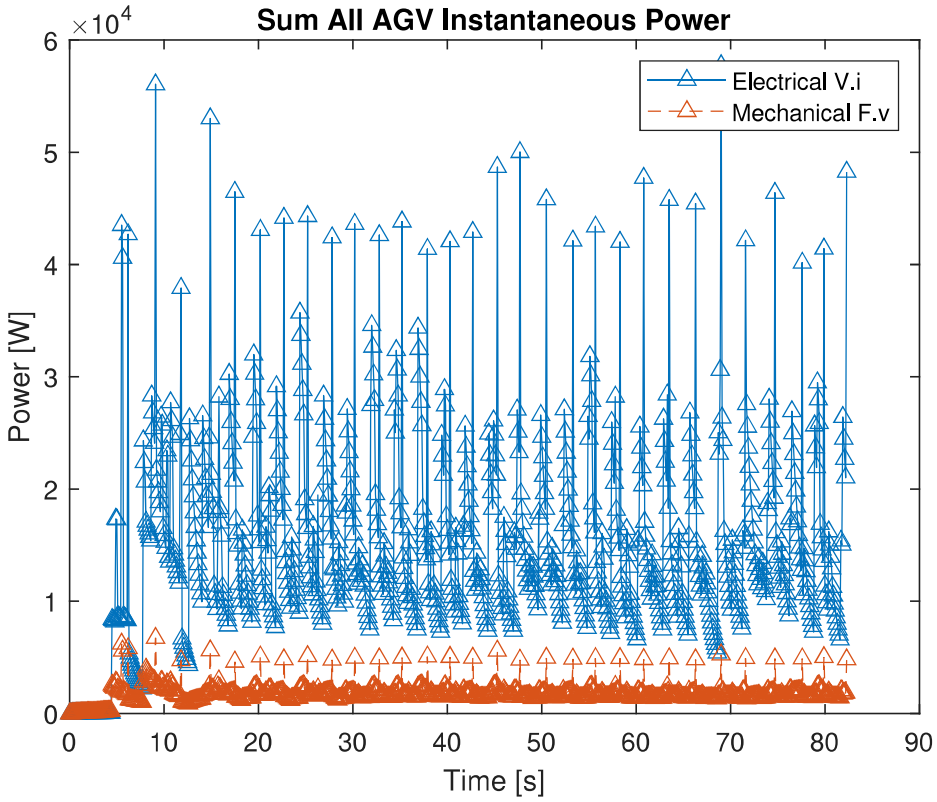

Figure 4.18: The power dissipated over time for all AGVs in total under Semaphore control.

4.7 Literature Review

Studies on the theoretical capacity of signalized intersections and roundabouts with an equivalent footprint indicate that in most cases, if there are few approach lanes small roundabouts will tend to have higher capacity. If there are many approach lanes signals tend to be more effective, unless the traffic on different approaches is extremely unequal [20].

A systematic procedure computing the conflict points in an intersection is given in [21]. Roundabouts tend to have a large number of merging and diverging conflicts, but fewer or none of the crossing and head-on conflicts which lead to the most serious collisions due to high relative speeds.

Intersection control often addresses crossing conflicts by separating vehicles in time, while they all take the shortest path straight through the intersection in the same way as if it was signal controlled. There are a wide range of optimal and heuristic approaches to solve for the speed profile, both decentralized and centralized, a good review is given in [22]. Many studies have looked at how to incorporate a proportion of human controlled vehicles which are not able to communicate their intention. One way of doing this is using traffic signals which only apply to human drivers [23]. The downside is that the nature of the intersection must remain similar to a traffic-light controlled one if non-communicating participants are going to be controlled by lights.

Recently a number of studies have extended intersection coordination of Connected and Autonomous Vehicles (CAVs) to the resolve the type of merging of diverging conflicts which occur and roundabouts. These are reviewed in [22]. A centralized solution with an intersection manager minimizing delay and energy consumption is described in [24]. This shows that a high proportion of vehicles need to be communicating for significant benefits to be realized.

A decentralized approach based on intent communication by way of virtual vehicles, can also be applied to roundabouts. In [25], reactive heuristics are shown to lead to poor performance compared to a model predictive control approach. The virtual vehicle concept allows common lane based heuristics such as car following to be extended to resolve conflicts in [26]. Another work investigating virtual lanes is [27]. Here a conflict graph is used to assign approaching vehicles to appropriate virtual lanes and a distributed controller is presented to stabilize the platoon.

Another approach presented in [28] is a decentralized solution to the global problem

of minimizing the delay. Proofs of completeness and optimality of the aggregate problem are given, making this technique very impressive. It is not shown to be applicable to roundabouts in any of the numerical examples, although the incorporation of optimal trajectory planning by the low level controller to execute merging makes it a good example of the combination of path planning and intersection management. Collision constraints are based on a conflict zone rather than conflict points as in [29]. The location of the conflict points is fixed by the fixed paths between the entry and exit lanes of the junction. The space inefficiency of the zone representation for multiple lanes is addressed by using multiple zones, one for each pair of lanes. The use of simultaneous path optimization might be expected to increase computational complexity and thereby reduce the number of vehicles with can be routed, however an attached video showing many vehicles interacting for about 10 minutes seems to refute this. It seem the ordering problem is resolved in a decentralized way based on game theory and the game 'Chicken.' Using game theory to resolve the ordering problem may give this approach an edge over the mixed integer optimization used in [29], in terms of how many vehicles they can control before running into execution time limits. It is a little surprising that the game would always produce the optimal ordering given the motion model used by each AGV. The consensus mechanism will be important here. Questions remain about the possibility of AGVs disagreeing about the order they calculate from the communicated position and speed data.

A similar method which solves the ordering ordering problem sequentially, followed by individual optimization of the approach speed along fixed paths is described in [30]. This method claims only local (per-vehicle) optimality for the speed choice sub problem, and makes it clear the crossing order at convergence will be suboptimal, and depends strongly on the decision order. The sub problem is posed as a Linear Quadratic Regulator, commonly seen in optimal control problems. In general terms, those early in the decision order will deviate from the plans less. This is more of a problem when vehicles are not uniform, as to reduce energy consumption a late arriving lorry should deviate as little as possible. A heuristic is given for the decision order based on the time to conflict arrival.

The use of optimal control in [30] is shared with many earlier works regarding co-ordination of Unmanned Arial Vehicles, many of which relax the assumption of static paths. In this way [31] addressed the full multi-vehicle motion planning problem for

small numbers of aircraft with simple dynamics. The craft were assumed to be differentially flat: that is, able to actuate in any of the workspace degrees of freedom independently, like a quadrotor. They were represented using bounding rectangles, leading to a slightly conservative mixed integer problem. The integer variables are used to choose which constraints are active. This might seem excessive when representing static obstacles, however when the constraints arise from other moving vehicles, the integer variables are a natural way to represent the passing-order problem. The scaling to larger numbers of vehicles is a particular challenge, due to the combinatorial explosion of possibilities.

An alternative approach to the coordination of differentially flat aircraft which uses a sequential solution of per-vehicle receding horizon sub problems to approximate the global solution is given in [32]. An earlier theoretical treatment based on iterative bargaining with soft collision constraints is given by [33]. The parameters are real numbers, and the constraints linear while the cost is quadratic. It may converge to an infeasible solution given a particular minimum safety distance even from a valid set of starting positions and speeds, and the suggested solution is to reduce the threshold until it becomes feasible.

More recently, solutions based on Distributed Model Predictive (DMPC) control have been developed. In [34], per-vehicle optimizations runs simultaneously to reduce execution time. This ensures recursive feasibility and closed loop stability. Another DMPC approach is given by [35]. This scales up to 25 vehicles in real time. the quadrotors concerned are all identical and differentially flat. For an under-actuated system like an AGV, some of the simplifications may no longer be possible.

4.8 Application Context

4.8.1 Roadmap-based AGV System

Consider a demand responsive AGV system for intra-logistics [**?**] or a smart factory [61]. The system is concerned with completing a series of material transfer tasks. A well known solution to motion planning in a well known environment involves simplifying the free space into a (possibly irregular) lattice of reachable states, connected by arcs if there exists a feasible transition from one state to the other, to create roadmap which can be encoded as a graph. A sequence of intermediate positions associated with each arc is sometimes stored alongside to avoid online re-computation. Using the roadmap graph, motion plans between any two states can be generated using a shortest path algorithm, which are detailed enough to be followed by the lateral position controller on board the vehicle.

In a centralized system the transfer tasks are assigned to available AGVs by a single scheduler which is aware of the status of every task and the position of every vehicle. The optimal assignment would minimize the makespan or total time for the completion of all tasks, but in practice this may be too time consuming, especially if new tasks are being generated all the time like in a fulfilment centre [**?**]. Conflict-free route planning depends on the task assignment and can be solved for jointly along with the assignment or performed sequentially based on a fixed assignment by searching the space time extended network to guarantee collisions are avoided.

Recently a number of decentralized systems have been developed which offer advantages in the number of vehicles that can operate in one area, reduced downtime for reconfiguration and safe interaction with human operators [**?**] . In [6], a roadmap representation is still used, but the roadmap is shared between vehicles. The partially decentralized system described in [**?**] combines traffic routing with per-intersection control is primarily roadmap based. In [56] is it improved with the possibility for an AGV to deviate from the roadmap based on its own sensors and based on a shared sensor state called the global live view. In such a decentralized system, an intersection controller cannot be assumed to know the motion plan of approaching vehicles, unless they communicate their intention as part of the protocol. To this end it is assumed a channel exists with sufficient bandwidth and a fixed latency T for the messages described in Section 4.9.1.

4.9 Modelling Plant and Interacting Digital Control Systems

To examine the approach to intersection control, we include an agent based model for every AGV at the intersection with access to strictly limited information which might be available from on-board sensors. All additional state information is sent according to the messaging interface defined in Section 4.9.1. It is an implementation of AIM* [29], with some adaptations to use the roadmap representation of Digani et al, which is typical in the AGV space [37]. AIM* was selected as it offers scope for the intersection controller to improve performance through optimization, compared to earlier interface descriptions such as [**?**].

4.9.1 Dual Waypoint Interface

The dual waypoint interface is designed to be decoupled from the algorithms for scheduling and routing as far as possible. In order to support decentralized routing with adaptive paths, each approaching vehicle must send an ApproachPlan message to containing a detailed plan for how it intends to cross the intersection. The ApproachPlan contains four parameters $d = [t_A, \boldsymbol{X}(s_A), v_A, \boldsymbol{X}(s)]$. The plan consists of a transmission timestamp t_A, a measured position $\boldsymbol{X}(s_A)$, and speed v_A at the given time and a sequence of feasible positions with no timing information, the path $\boldsymbol{X}(s)$.

Embedding the path in each request for guidance means that approaching AGV can use obstacle avoidance planning before they enter the approach lane, and still receive the correct speeds at the intersection. As a result the size and shape of the conflict zone is not fixed but depends on the current traffic situation and the approach plans received.

The conflict zone shape is calculated by discretizing $\boldsymbol{X}(s)$ into linear segments of length $L = 1\text{m}$ and searching for points where the minimum distance between two segments exceeds the diameter of the AGV bounding circle, and the direction of the segment is different. This ensures there is no conflict point identified where one segment joins another, which arises when two AGV are following the same path one after the other.

The intersection controller is responsible for generating an optimal speed profile for this path $v(t)$, to create a trajectory which satisfies the collision avoidance constraints

with the trajectories of all known approaching vehicles $\boldsymbol{\xi}_i(t) \forall i \in N$.

The trajectory across the intersection $\boldsymbol{\xi}(t)$ is found from the path $\boldsymbol{X}(s)$, the start time t_A and start position $\boldsymbol{X}(s_A)$ using Equation 4.33.

$$\boldsymbol{\xi}(t) = \boldsymbol{X}(s_A) + \int_{t_A}^{t_L} \boldsymbol{X}\left(v(t)\right) dt \tag{4.33}$$

The speed profile is always expressed as two average speeds for two segments. The first segment AB begins at the position of the AGV at transmission time $\boldsymbol{X}(s_A)$, and ends at the nearest edge of the intersection conflict zone $\boldsymbol{X}(s_B)$. The second segment BC begins at $\boldsymbol{X}(s_B)$ and ends at the far edge of the intersection conflict zone $\boldsymbol{X}(s_C)$.

To represent this level of detail, the DualWaypoint contains four parameters $d = [t_B, t_C, s_B, s_C]$. These are independent of the discretization in the ApproachPlan, and expressed in path coordinates. The flow of messages over time is shown in Figure 4.19.

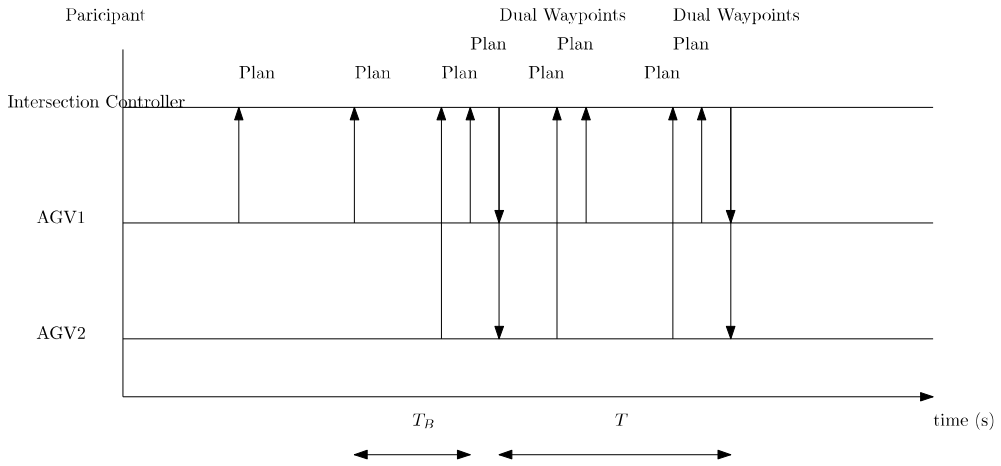

Figure 4.19: Sequence diagram for two AGVs communicating with the Intersection Controller which sends a DualWaypoint message to every known AGV every T seconds, considering the latest ApproachPlans it has received to date.

4.9.2 Longitudinal Speed Control

Longitudinal Speed Control for each Individual AGV is based on two main behaviours. The first one determines the speed on unconflicted links. The second one is required meet the timing specification contained in the Dual Waypoint message, subject to disturbances and uncertainty in the plant using position feedback.

Previous authors have modelled the speed on unconflicted links using car-following behaviour models. Automated traffic is assumed to follow an Adaptive Cruise Control Model with set headway, while human operated vehicles follow the Intelligent Driver Model in [**?**]. In the AGV space it is common to simplify car-following with mutual exclusion of discretized roadmap segments [**?**] so we follow this scheme for the main results. Some results with mutual exclusion turned off are given in in Section 4.6.1 before the main results with mutual exclusion in Section 4.6.2. The update period $T_L = 0.1$s must shorter or equal to that of the intersection controller T.

The Dual Waypoint Timing Specification is met with a constant acceleration model based on the collision-free operation modes in [44]. Our simulation incorporates two modes, depending on whether the vehicles position feedback $\boldsymbol{X}(\hat{s})$ at time $\hat{t}$ indicates it is approaching the conflict zone so $\hat{s} < s_B$ or already inside it so $s_B \leq s_A < s_C$. If the AGV has passed the conflict $\hat{s} > s_C$ then its speed is unconstrained from the perspective of this intersection controller. In the simulation exiting vehicles would accelerate to maximum speed, at α_{max}.

On approach to the conflict zone, where $\hat{s} < s_B$, the approach acceleration α_{AB} is given by Equation 4.34.

$$\alpha_{AB} = \frac{(s_B - \hat{s}) - \hat{u}(t_B - \hat{t})}{0.5(t_B - \hat{t})^2} \tag{4.34}$$

Within the conflict zone $s_B \leq s_A < s_C$ the acceleration α_{BC} is given by Equation 4.35.

$$\alpha_{BC} = \frac{(s_C - \hat{s}) - \hat{u}(t_C - \hat{t})}{0.5(t_C - \hat{t})^2} \tag{4.35}$$

4.9.3 AGV Motor Dynamic and Electrical Model

For the dynamics, every AGV was assumed to have the same mass $M = 100$kg whether loaded or unloaded, reflecting a negligible cargo mass, for example spare parts for mobile phone repair. An AGV may be propelled by brushless DC motors, which provide high torque and efficiency. Even so, a major source of power loss is internal resistance of the windings and magnetic losses in the core. The field strength of the magnets, the number of poles and the number turns of the armature coils can be captured in the motor constant k_T relating torque τ [Nm] to armature current.

$$\tau = k_T I_a \tag{4.36}$$

Similarly, the rotational speed ω [rpm] is related to the back emf ϵ [V] by Equation 4.37.

$$\omega = k_e \epsilon_D \tag{4.37}$$

These can be combined to give the plant model for one AGV in Equation 4.38

$$\ddot{x} = \frac{u \cdot k_T (V_{CC} - \epsilon_D)}{M R_a d_W / 2} \tag{4.38}$$

There are numerous loss sources in an electric motor such as winding resistance, flux leakage, eddy currents in the core and so on [40]. By using real-world measured mechanical power output and electrical input, an equivalent winding resistance R_a for the simple model can be found. The parameters are shown in Table 4.6.

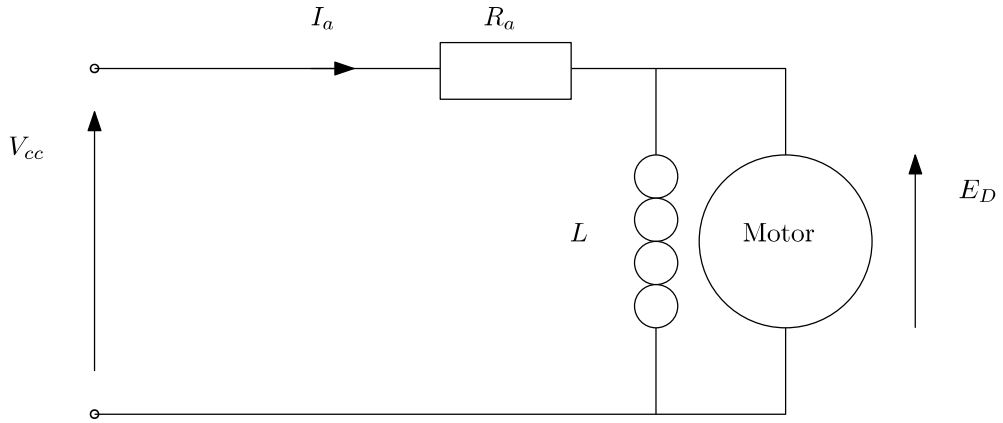

Figure 4.20: Steady state equivalent circuit for a DC motor.

As the top speed $v = 5\text{m/s}$ is quite low, and the vehicles stop and start frequently, air resistance which varies according to Equation 4.39 was found to be an order of magnitude smaller than the electrical losses, based on a frontal area $A = 1\text{m}^2$ and the The drag coefficient C=1 for a cuboid shape was used, taken from [41]. Air density is taken to be $\rho = 1.224\text{kg/m}^2$.

$$F_a = C \rho A v^2 \tag{4.39}$$

A brushless DC motor for an industrial vehicle typically has a constant voltage from a battery pack [**?**]. In this case we set V_a =72V, within the range tested in [38]. Torque can be varied from zero to maximum by changing the slip angle between

Table 4.6: Motor parameters used in simulation. Electric fork lift mass and speed [**?**]. Motor and Electrical parameter from [38]. *Computed for equivalent circuit in Equation 4.38 to match τ_{max} at i_{max}

a_{max}	2.5	m/s^2
v_{max}	5.0	m/s
k_v	6	rpm/V
k_T	1.53	Nm/A
P_{mech}@375rpm	3.6	kW
P_{elec}@375rpm	6.37	kW
τ_{max}	127.2	Nm
$*R_a$	0.5	Ohms
V_{CC}	72	V
i_{max}	80	A
M	400	kg
d_W	0.256	m

the magnetic field generated digitally by the three phase coils and the magnetic field generated by the hgh strength magnets fixed on the rotor. The output of the vehicle's longitudinal speed controller must therefore be a duty cycle $-1.0 < d < 1.0$. A value of zero corresponds to zero torque, where the slip angle is zero, and a value of +/-1.0 to a slip angle of +/-90 degrees where torque is at a maximum in forward or reverse respectively.

4.10 Method

To examine control of variable numbers of vehicles, and comment on safety effects as well as performance of experimental algorithms utilizing a dynamic simulation test environment seemed prudent to start with. The goal here is to examine edge cases which are safety critical (may lead to a collision between AGV). The different algorithms will be compared based on execution time, total travel time, total energy consumption and mean throughput delay with a given traffic pattern.

4.10.1 Conflict Zone Approximation

The collision avoidance constraints are simplified by merging all the conflicts on each path, to keep only the smallest s_B and the largest s_C for that path. The extent of the conflict zone for vehicle$i$ is given by Equation 4.40. The conflicted segment of each vehicle's path lies between s_B the smallest value of s which satisfies Equation 4.40 and s_C the largest value. The union of these conflicted segments form the total conflict zone, which is an irregular non-convex, connected compound shape.

In some cases it may be advantageous to limit mutual exclusion by only considering path segments which have different orientations to be in conflict, even if they satisfy Equation 4.40. This could allow closer spacing of AGV travelling in the same direction by following according to the distance measured to leader by on-board sensors.

$$||X_i(s) - X_j(t)|| < W_v \quad \forall j \in N, \quad j > i \tag{4.40}$$

4.10.2 Objective

The objective to minimize the total travel time is given by Equation 4.41. It is linear terms of the reciprocal speed vector $\boldsymbol{\phi} \in R^{(n \times n)}$, which has up to two elements per AGV. One for the approach if it has not yet been passed and one for the conflict so $\boldsymbol{\phi}_i = [\phi_{AB}, \phi_{BC}]$. The segment lengths for the approach and the conflict are contained in distance vector $\boldsymbol{d}$ so $\boldsymbol{d}_i = [d_{AB}, d_{BC}]$.

$$\begin{gathered} \min_{\boldsymbol{\phi}} \boldsymbol{J}_T = \boldsymbol{d}^T \boldsymbol{\phi} \\ \text{subject to} \\ \boldsymbol{\phi} > \phi_{min} \\ \boldsymbol{\phi}^T \boldsymbol{H_{ij}} \boldsymbol{\phi} > 0 \quad \forall i, j \in [1, p] \quad \text{with } j > i \end{gathered} \tag{4.41}$$

The condition $j > i$ in Equation 4.41 indicates that the number of constraints varies with the number of vehicles p as $\frac{p(p-1)}{2}$. This corresponds to one constraint between each pair of approaching AGVs.

4.10.3 Differential Constraints

Vehicle acceleration limits are dealt with implicitly, by the maximum speed which can be expressed as a lower bound on $\boldsymbol{\phi} < \phi_{min}$. The simulated value ϕ_{min}=5m/s, is

reachable within a certain distance d_{min} from any feasible starting speed, assuming a constant limited acceleration a_{max} according to $v_{max} = \sqrt{2a_{max}d_{min}}$. Using the parameters from Table 4.11, the acceptable distance is d_{min} =5m.

4.10.4 Online Feedback Considerations

In order to guarantee feasibility we need only to ensure the conflict zone length d_{BC} and the approach length d_{AB} are both greater than d_min when vehicles receive their instructions. A vehicle proceeding toward the conflict will eventually pass the point of no return where $d_{AB} = d_{min}$. It can not be guaranteed that any instructions sent after this point can be satisfied by the on-board longitudinal control. Any vehicle past the point of no return appears in the optimization as a constant constraint on the speeds of subsequent vehicles. The constraint uses the latest reported speed and position for real-time feedback, so if a vehicle past the point of no return fails to meet its deadline, the later vehicles can be safely delayed until it leaves the conflict zone.

4.10.5 Conflict-zone Collision Avoidance Constraints

By definition, each intersection controller is responsible for one conflict zone, constructed as explained in Section 4.10.1. This makes it possible to express the constraint that vehicles do not collide in terms of time. Vehicle i arrives at the first conflicted segment $\omega_m in$ and departs from the last at ω_{max} . The following three subsections set out three alternative ways of expressing the collision avoidance constraints which have been evaluated. The arrival time is given by Equation 4.42. Considering average speeds, the departure time ω_{max} is also linear, this is given by Equation 4.44.

$$\omega_i^{min} = d_{AB}\phi_{AB} = \boldsymbol{e}^T\boldsymbol{\phi}_i \tag{4.42}$$

Where

$$\boldsymbol{e}^T = [d_{AB}, 0] \tag{4.43}$$

and

$$\omega_i^{max} = [d_{AB}, d_{BC}] \begin{bmatrix} \phi_{AB} \\ \phi_{BC} \end{bmatrix} = \boldsymbol{f}^T\boldsymbol{\phi}_i \tag{4.44}$$

Where

$$\boldsymbol{f}^T = \begin{cases} [d_{AB}, d_{BC}], & \text{if } d_{AB} > 0 \\ [0, d_{BC}], & \text{otherwise} \end{cases} \tag{4.45}$$

Following [37], the time window between ω_{min} and ω_{max} may be expressed in terms of the midpoint α and the extent β. In this way the collision avoidance constraints in Equation 4.46 are independent of the order in which AGV i and AGV j arrive.

$$|\alpha_i - \alpha_j| > \beta_i + \beta_j \tag{4.46}$$

Here

$$\alpha_i = \omega_i^{max} + \omega_i^{min} \tag{4.47}$$

represents the midpoint of the time vehicle i occupies the conflicted segment and and

$$\beta_i = \omega_i^{max} - \omega_i^{min} \tag{4.48}$$

represents the range of the time either side of the midpoint, both scaled by a factor of two.

In matrix form this can be written

$$\alpha_i = \boldsymbol{f}^T\boldsymbol{\phi}_i + \boldsymbol{e}^T\boldsymbol{\phi}_i = \mathbf{1}_i^T\boldsymbol{A}\boldsymbol{\phi}_i \tag{4.49}$$

with $\boldsymbol{A} = \text{diag}(\boldsymbol{f} + \boldsymbol{e})$

$$\beta_i = \boldsymbol{f}^T\boldsymbol{\phi}_i - \boldsymbol{e}^T\boldsymbol{\phi}_i = 1_i^T\boldsymbol{B}\boldsymbol{\phi}_i \tag{4.50}$$

with $\boldsymbol{B} = \text{diag}(\boldsymbol{f} - \boldsymbol{e})$

4.10.6 Extra Constraint Between Vehicles in the Same Lane

Vehicles travelling in the same lane forming a moving queue are more constrained than vehicles approaching a conflict zone. AGV are assumed here to be unable to overtake safely based on local sensors. This is likely to hold even with recent AGV which are capable of significant autonomy including adaptive path planning. This is because floor space is at a premium in a logistic environment so the gaps between the shelves are unlikely to be much wider than one AGV.

Indices are increasing so vehicle (i+1) is following behind vehicle (i). The safety constraint between vehicles in the same lane l to ensure they remain a safe distance L apart is given by Equation 4.51

$$s_i > s_{i+1} + L \quad \forall i \in l \tag{4.51}$$

This can be expressed in terms of minimum time to collision of $TTC_{min} = 2L/(v_i+v_{i+1})$ as in Equation 4.52.

$$(s_i - s_{i+1})/(v_i - v_{i+1})) > TTC_{min} \tag{4.52}$$

It is a little awkward to capture this constraint exactly using the average speed on two segments. This approximation in Equation 4.53 was tested.

$$(s_i - s_{i+1})(\phi_i - \phi_{i+1})) > TTC_{min} \tag{4.53}$$

4.10.7 Non-Convex Quadratic Constraints Optimal Intersection Control

Equation 4.46 can be converted to standard form by squaring both sides and substituting the matrix expressions for α_i and β_i. This gives the matrix inequality for each pair of vehicles shown in Equation 4.54.

$$[\boldsymbol{\phi}_i^T, \boldsymbol{\phi}_j^T] \begin{bmatrix} \boldsymbol{\Lambda}_{ij}^{ii} & \boldsymbol{\Lambda}_{ij}^{ij} \\ \boldsymbol{\Lambda}_{ij}^{ji} & \boldsymbol{\Lambda}_{ij}^{jj} \end{bmatrix} > 0 \tag{4.54}$$

The four pairwise submatrices can be expressed in terms of the diagonalized distance $\boldsymbol{A}$ and $\boldsymbol{B}$ as follows:

$$\boldsymbol{\Lambda}_{ij}^{ii} = (\boldsymbol{A}_i - \boldsymbol{B}_i)\mathbf{1}_i\mathbf{1}_i^T(\boldsymbol{A}_i + \boldsymbol{B}_i) \tag{4.55}$$

$$\boldsymbol{\Lambda}_{ij}^{jj} = -(\boldsymbol{A}_j + \boldsymbol{B}_j)\mathbf{1}_j\mathbf{1}_j^T(\boldsymbol{A}_j + \boldsymbol{B}_j) \tag{4.56}$$

$$\boldsymbol{\Lambda}_{ij}^{ij} = \boldsymbol{\Lambda}_{ij}^{jiT} = -(\boldsymbol{A}_j + \boldsymbol{B}_j)\mathbf{1}_j\mathbf{1}_i^T(\boldsymbol{A}_i + \boldsymbol{B}_i) \tag{4.57}$$

For more than two vehicles this can be arranged into a block diagonal matrix $\boldsymbol{H}_{ij} \in R^{(n\times n)}$ which is compatible with the input parameters, but still only represents the constraints between a pair with zeros for the other elements. The full constraint matrix $\boldsymbol{H}$ is the sum of these pairwise matrices, for every pair with $j < i$.

4.10.8 First-Come-First-Served Optimal Linear Intersection Control

With a fixed ordering such as First-Come-First-Served, the reciprocal speed vector $\boldsymbol{\phi}$ is arranged in arrival order.

The constraint in Equation 4.46 only needs to be applied between adjacent vehicles and it will hold for all vehicles. This reduces the number of constraints between n vehicles to $n-1$.

The timing constraint that the leader exits the conflict zone before the follower enters is

$$\omega_i^{max} > \omega_{i+1}^{min} \tag{4.58}$$

This can be expressed as

$$\boldsymbol{e}_i^T \boldsymbol{\phi}_i > \boldsymbol{f}_{i+1}^T \boldsymbol{\phi}_{i+1} \tag{4.59}$$

leading to a pairwise matrix $Q^{ij} \in R^{(n \times n)}$

$$\boldsymbol{Q}^{ij}\boldsymbol{\phi} = \begin{bmatrix} 0 & \dots & & \\ \dots & \boldsymbol{e}_i^T & -\boldsymbol{f}_{i+1}^T & \dots \\ & & \dots & 0 \end{bmatrix} \begin{bmatrix} \vdots \\ \boldsymbol{\phi}_i \\ \boldsymbol{\phi}_{i+1} \\ \vdots \end{bmatrix} \tag{4.60}$$

The pairwise Q^{ij} matrices are added together to get A_{ub} in Equation 4.61,

$$A_{ub}\boldsymbol{\phi} > 0 \tag{4.61}$$

The vehicles past the point of no return with latest feedback reciprocal speeds for each incomplete segment

$$\boldsymbol{p}_k = [1/v_{AB}, 1/v_{BC}] \tag{4.62}$$

are included in Equation 4.63

$$\boldsymbol{e}^T \boldsymbol{\phi}_i > \boldsymbol{f}^T \boldsymbol{p}_k \tag{4.63}$$

here $\boldsymbol{f}$ defined in Equation 4.45.

4.10.9 Semaphore Based Collision Avoidance

The constraints can be enforced without any optimization using a common synchronization object the binary semaphore. This is also based on first come-first-served ordering and the intersection controller gives the semaphore to the closest vehicle who provides an Approach Plan. This requires special messages in the dual waypoint interface, as the intersection controller makes no attempt to predict the time the conflict will become free. It issues a full speed ahead command to the vehicle with the semaphore

and a space exclusion to all other vehicles. This consists of a distance along the AGVs submitted plan which it is not allowed to pass until given further instructions.

This type of system is expected to lead to sub optimal throughput but be fast to calculate and guarantees safe operation. Similar schemes have been described in the literature so it is included in the comparison to give an idea of the benefits of departure time modelling and approach speed synchronization.

4.11 Numerical Results

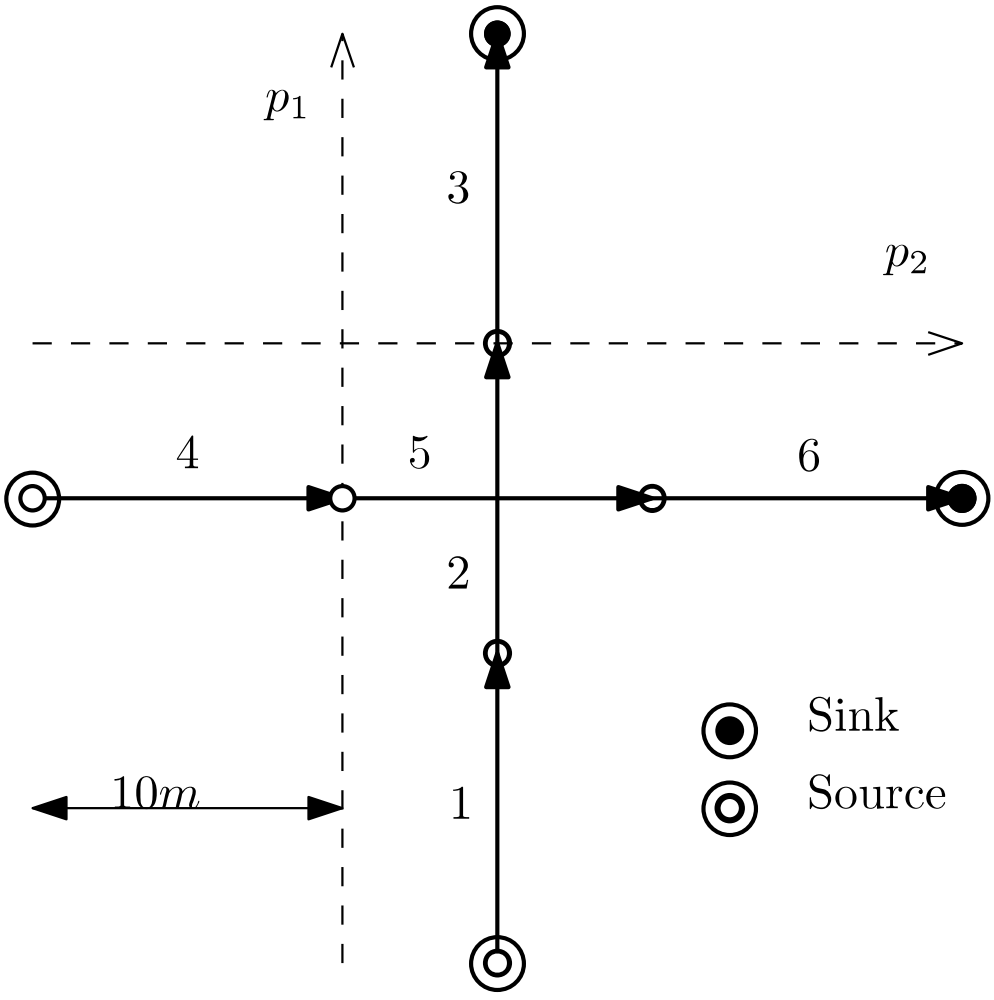

Figure 4.21: Intersection layout with two conflicting routes.

The different approaches to intersection control were evaluated on a simulation of

	λ_1	λ_2	f
HLHT	0.5	0.5	2
HLMT	0.1	0.5	2
HLLT	0.1	0.1	2
LLHT	0.5	0.5	10
LLMT	0.1	0.5	10
LLLT	0.1	0.1	10

Table 4.7: Parameters for test scenarios. All units s$^-$1.

	T[s]	TTT[s]	t[s]	Δ[s]	Ee[MJ]	Em[MJ]	Ex T[s]
FIFO	45.7	181.0	6.033	0.033	43.906	30.323	0.0036
Quad	44.8	181.6	6.0533	0.053	44.832	30.964	0.5252
Sema	83.9	326.4	10.88	4.88	159.1	68.140	¡0.001

Table 4.8: Intersection performance over 30 crossings with three different controllers for the HLHT scenario.

a simple intersection, comprised of two 30m lanes which cross in the middle as shown in Figure 4.21. There are two entrances to the map, one at the start of each lane. By varying the arrival rate λ and the update frequency f, six scenarios were created with the parameters shown in Table 4.11.

Each scenario is identified with the first two characters relating to the latency between periodic messages from the intersection controller where High Latency is 500ms and Low Latency is 100ms and the second two relating to the arrival rate, where High Traffic has λ=10 arrivals per second on both approaches, Low Traffic has λ=2 arrivals per second on both, and Mixed Traffic has one lane with λ_1=10 and the other with λ_2=2. For example High Latency, High Traffic becomes HLHT

The effects of the different controllers can be seen in the position time trace for 30 simulated crossings. The conflict zone is protects the intersection between the two lanes at s = 15m. Both lanes are collapsed onto one diagram, with $\times$ markers for vehicles travelling along the x axis and $\triangle$ markers for vehicles travelling along the y-axis. The controller is successful provided only one type of marker is present in the conflict zone at one time. All controller are safe, so the main comparison is how much the vehicles must slow down, shown by the gradient of the lines. The benefit of modelling the

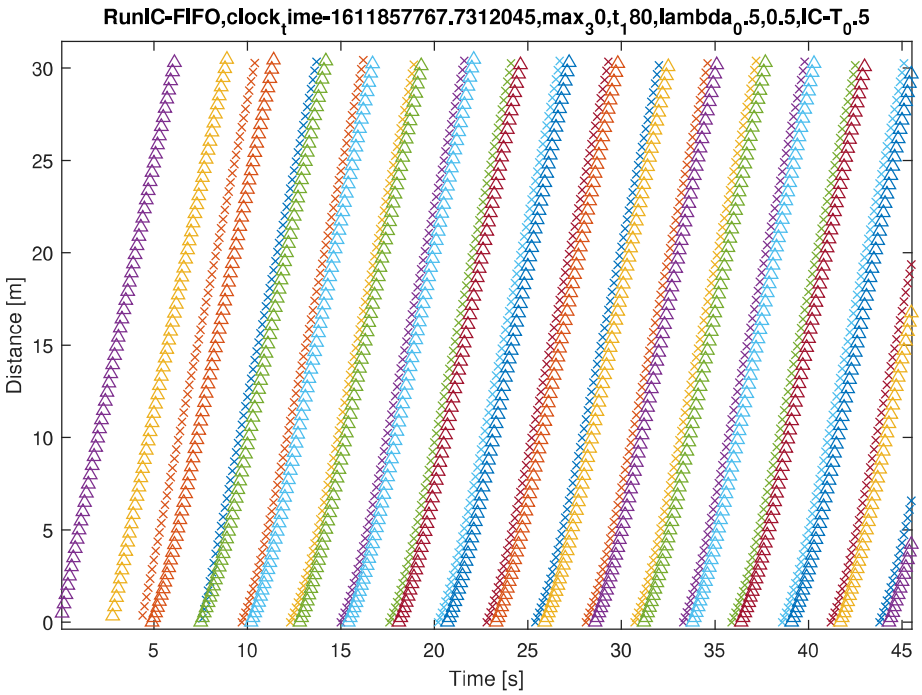

Figure 4.22: Position-Time trace for HLHT Scenario under FIFO controller

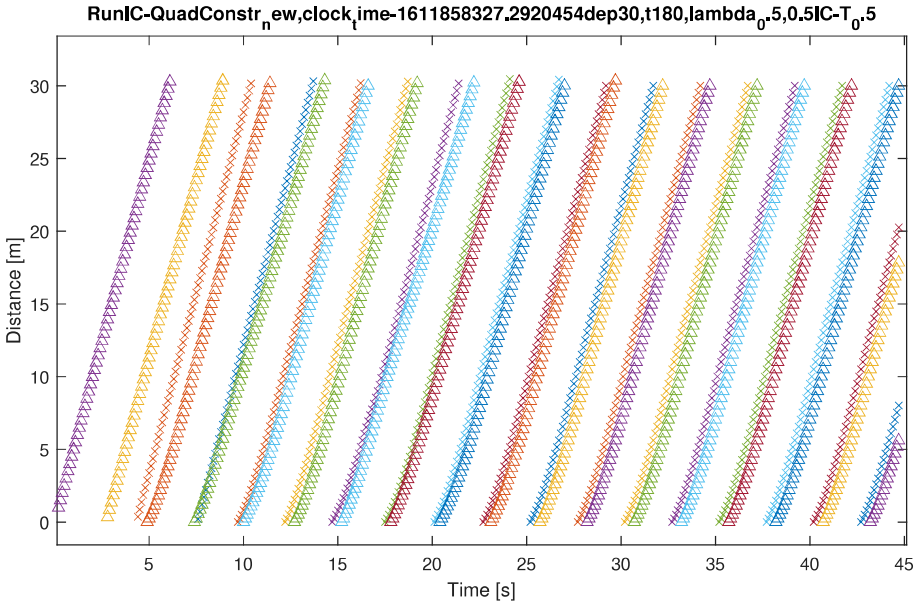

Figure 4.23: Position-Time trace for HLHT Scenario under Quadratic Constraints controller

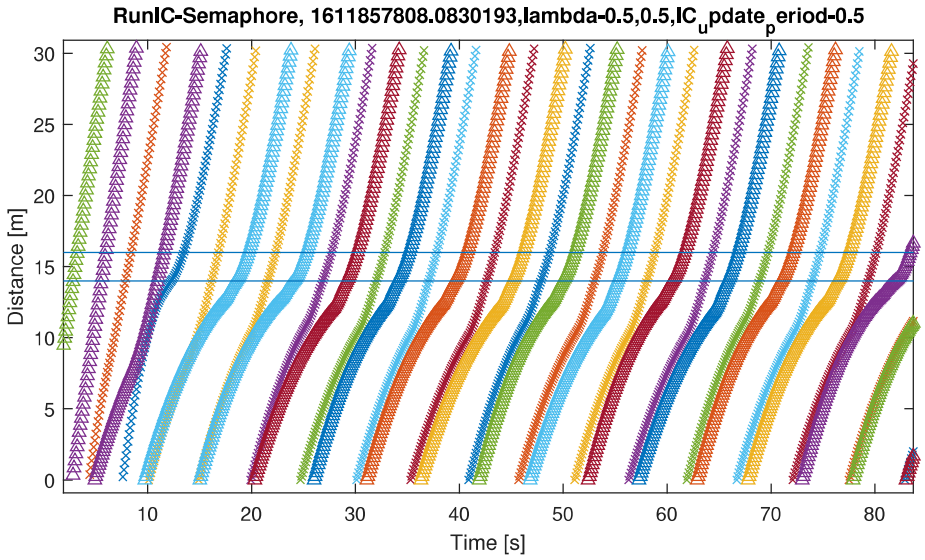

Figure 4.24: Position-Time trace for HLHT Scenario under Semaphore controller

departure time and adjusting speeds in advance is clear from comparing the optimal controllers in Figure 4.11 and Figure 4.11 with the Semaphore approach in Figure 4.11. This corresponds to a reduction in delay of 4.85 seconds per vehicle according to Table 4.11.

The two optimal methods are very close, with FIFO achieving a slight improvement in total travel time of 0.6 seconds, but a lower completion time by 0.9 seconds. This discrepancy may occur because the waiting time in the arrival queue is not counted in the total travel time, which should be addressed in further testing. It is more likely the Quadratic constraints achieved a slight improvement in throughput because of the freedom to vary the departure order. However, the departure order in Figure 4.11 turns out to be close to FIFO anyway.

Another avenue of comparison is the energy usage. The semaphore method uses much more energy as the vehicles have to slow down more. Energy usage is not included in the objective for the optimal methods, so the question depends on whether higher average speeds or more acceleration lead to higher losses with our simple motor model.

The power consumption increase due to acceleration clearly dominates in Figure 4.11, as the mechanical power is around 50 percent greater than in either of the optimal runs. This difference is compounded by the reduction in motor efficiency in high acceleration so the resultant increase in electrical poser dissipation is much greater, closer to 200 percent.

There is still little to distinguish the two optimal approaches. Although unlike the delay, in this case maintaining FIFO order leads to a slight improvement: 43.9 MJ total energy compared to 44.8MJ. A spike in usage at around 18 seconds can be seen in Figure 4.11, possibly this corresponds to a change in order which leads to lower delay but uses some extra energy.

4.11.1 Impact of Analytical Hessian on Execution Time of Trust Region Method

The optimization problem with quadratic constraints described in Section 4.10.7 was implemented in Python and solved periodically based on the latest position information at the specified control frequency f. The method chosen was 'trust-constr' from the Scipy.Optimize library [57]. Trust region methods make use of the exact

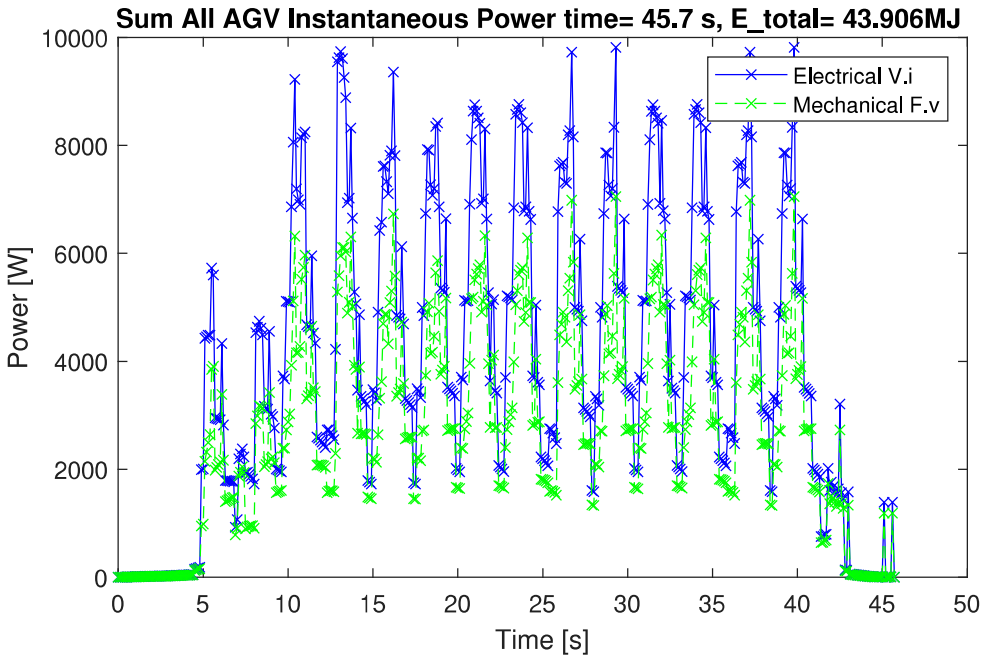

Figure 4.25: Power Dissipation-Time trace for HLHT Scenario under FIFO controller

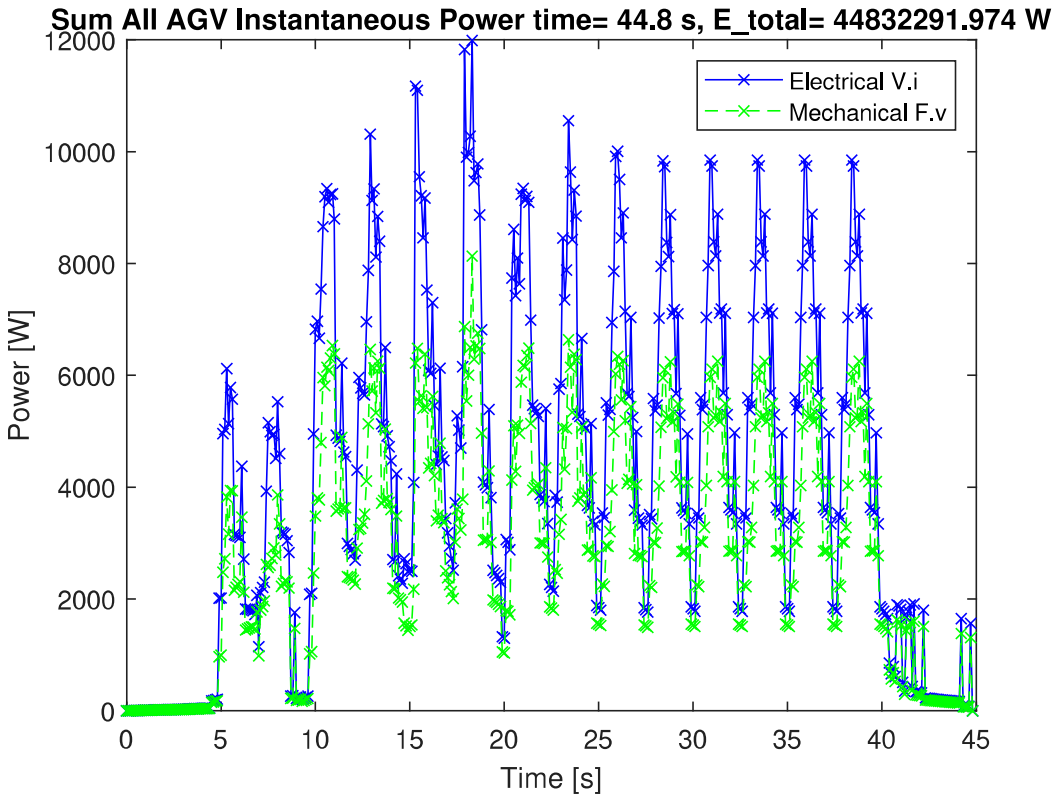

Figure 4.26: Power Dissipation-Time trace for HLHT Scenario under Quadratic Constraints controller

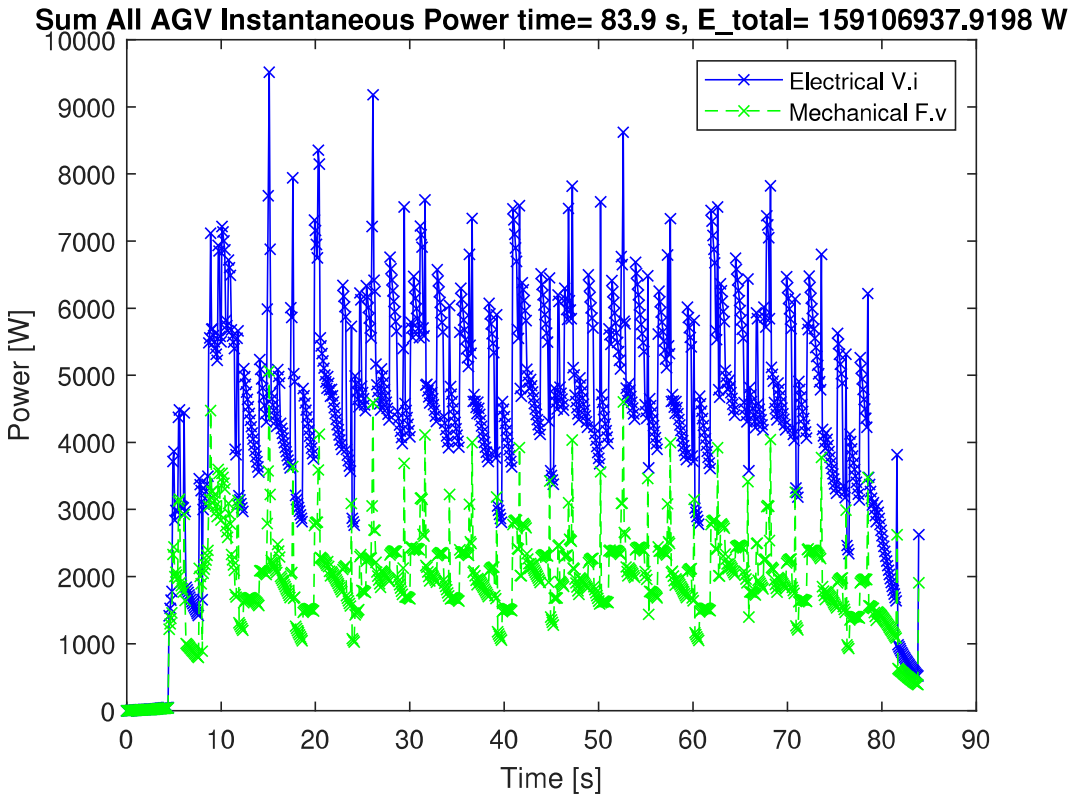

Figure 4.27: Power Dissipation-Time trace for HLHT Scenario under Semaphore controller

Semi-Definite Program relaxation for the Trust Region Sub-problem (TRS), of optimizing a non-convex quadratic objective subject to a Euclidean ball constraint, to iteratively solve general non-convex function with non-convex constraints by successive approximation[58]. They are likely to be more effective when the general problem has more in common with the TRS and recent methods have been proven to solve variants of that problem in linear time in terms of the input [**?**]. Unlike some other general constrained optimization methods in Scipy.Optimize such as SLSQP, 'trust-constr' can make use of the analytical Hessian for the objective and constraints which may be important to exploit the linear objective and quadratic constraints.

The Hessian must be provided to SciPy.Optimize in the form of a linear combination rather than a stacked matrix. This is to avoid forming the complete Hessian $H \in R^{(n \times np)}$ which may use a significant amount of memory for large problems. Instead, the analytical Hessian function must accept an additional parameter $v \in R^{(1 \times p)}$. This is a vector the same length as the constraints $c_{ineq} \in R^{(1 \times p)}$. The Hessian is returned as a $R^{(n \times n)}$, the weighted sum of pairwise blocks scaled according to $\sum_{i=1}^{p} v_i H_i j$.

With the analytical Hessian the average execution time for the Quadratic Constraints method over the HLHT run in which 30 vehicles passed through the intersection was 0.5251 seconds, varying between 0.0512 seconds to 1.215 seconds as the number of constraints varied from 1 to 6. Without the analytical Hessian of the constraints Without the analytical constraint Hessian the mean time taken over the same run was 0.383 seconds, varying between 0.0468 seconds to 7.696 seconds. It is surprising that the worst case time is so much worse and yet the mean time is better. This suggests that in the test data there are more cases with few constraints. It also motivates investigation into the cause of the outlier time.

The execution time with the FIFO controller never exceeds 15.6 milliseconds on the same set of problems, with the average being 3.6 milliseconds.

4.12 Conclusion

The advantages of centralized intersection optimization shown by previous authors are supported by our results. Furthermore we show that enforcing first-in-first-out ordering leads to very similar performance in both delay and energy consumption on a simple intersection comprising two crossed lanes. For this reason the FIFO controller is a promising choice for real world implementation, as it can be solved orders of magnitude

faster and captures almost all of the throughput advantage. The next step is to ensure this result holds for more complex intersections, where exploring alternative orderings may be more significant to the objective.