Noname manuscript No.
(will be inserted by the editor)

Optimal Control of AGV Intersections and the Avoidance of Queues

Edward Derek Lambert · Richard Romano · David Watling

Received: date / Accepted: date

Abstract Insert your abstract here. Include keywords, PACS and mathematical subject classification numbers as needed.

Keywords First keyword · Second keyword · More

1 Introduction

Reservation-based intersection management for preventing collisions between autonomous vehicles at intersections by V2V communication has been the topic of numerous studies considering road traffic [5]. Some studies utilized a First-Come-First-Served (FCFS) policy which was shown to outperform signal control in some situations [11]. Problematic cases where performance could be worse than signal control were identified by [**?**]. To capture the potential capacity improvements other works have used convex optimization [7].

2 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.

E.D. Lambert · R. Romano · D.P.Watling
Institute for Transport Studies, University of Leeds, 35- 41 Universiy Road, LS2 4JT
E-mail: tsedl@leeds.ac.uk E-mail: R.Romano@leeds.ac.uk E-mail: D.P.Watling@its.leeds.ac.uk

2.1 Dynamic Simulation

In order to comment on safety performance the simulated dynamics of the AGV must be more detailed than the model used by the intersection controller. These additional dynamic effects mean the timing specification sent by the intersection controller will always be missed by a small amount as it would be if the controller was communicating with embodied robots acting in a real logistics environment. In order to comment on the total energy consumption, it would be useful to model second order dynamics, allowing the dissipated power to be approximated by the acceleration. Acceleration is a good proxy for motor power in a low speed system like logistics, where there is little air resistance but lots of stopping and starting. In order to investigate the effect of communication latency the simulation time-step should be smaller than the hypothetical communication latency for client/server control over a limited bandwidth network. Lateral tracking behaviour need not be considered if we assume all AGV are able to traverse the given path with negligible lateral error. This is not too unrealistic if the paths account for dynamic limitations limitations of the vehicles which will traverse them such as radial polynomials [8] or clothoid transitions [17].

With these requirements in mind, a number of off-the-shelf traffic simulation software packages were investigated. To give a brief overview we considered SUMO [3], DRACULA [21], MATSim [14], PTV VISSIM [15] and Aimsun [20]. Each one lacked either the required time granularity, the second order dynamic or the ready ability to add custom vehicle controllers and intersection controllers, communicating with custom messages at a certain rate.

Another option considered was using an existing robot dynamic simulator. Stage [25] is designed for large

numbers of robots, but is unsuitable as it relies on a first order dynamics. Another option was Gazebo [23] which has been used in similar works such as [26] and [28], where ROS (Robot Operating System) libraries were used to implement the control algorithms, so the same binaries could equally be linked to a full Gazebo simulation or a collection of physical test robots. This is a great option but would force us to implement robust lateral control and other systems needed for a real AGV which are incidental to our contribution. ROS1 also has some issues with multiple vehicles which require another system to work around, for example another library Fawkes in one notable competition [22]. An better option would be the new ROS2 library which does away with the single master architecture and TCP messaging protocol [12] but is more sparsely documented and under active development at the time of writing, so we leave it for further work.

Initial experiments reported in [19] and [18], made use of a bespoke Python simulation. We continued to develop this custom simulation to add the required functionality, instead of heavily modifying an existing microsimulation package or adding multiple vehicles to a dynamic simulator like Gazebo. The Python files for our simulation, and all the tested controllers are available in GitHub [16].

2.2 Roadmap-based AGV System

In general, a demand responsive AGV system for intralogistics or a smart factory is concerned with completing a series of material transfer tasks. Each task consists of moving one unit load from a given pick location to the specified drop location. 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, and different heuristics such as nearest-first assignment are useful. 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. In [26], a roadmap representation is still used, but the roadmap is shared between vehicles. The partially decentralized system described in [9] combines traffic routing with per-intersection control is primarily roadmap based. In [4] 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 2.4.

2.3 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.

$$\tau = k_T I_a \tag{1}$$

Similarly, the rotational speed ω [rpm] is related to the back emf ϵ [V] by Equation 2.

$$\omega = k_e \epsilon_D \tag{2}$$

These can be combined to give the plant model for one AGV in Equation 3

$$\ddot{x} = \frac{u \cdot k_T (V_{CC} - \epsilon_D)}{M R_a d_W / 2} \tag{3}$$

There are numerous loss sources in an electric motor such as winding resistance, flux leakage, eddy currents

Table 1 Motor parameters used in simulation.*Computed for equivalent circuit in Equation eq:model 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

in the core and so on [24]. 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 1.

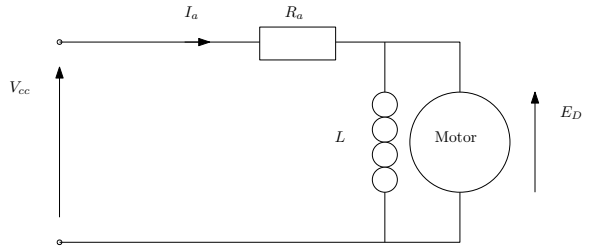

Fig. 1 Steady state equivalent circuit for a DC motor.

The

As the top speed $v = 5$m/s is quite low, and the vehicles stop and start frequently, air resistance which varies according to Equation 4 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 [**?**]. Air density is taken to be $\rho = 1.224\text{kg/m}^2$.

$$F_a = C\rho A v^2 \tag{4}$$

A brushless DC motor typically has a constant voltage from a battery pack, in this case we set V_a =72V, achievable with six golf cart batteries in series. Torque can be varied from zero to maximum by changing the slip angle between 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.

2.4 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 $L = 1$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 5.

$$\boldsymbol{\xi}(t) = \boldsymbol{X}(s_A) + \int_{t_A}^{t_L} \boldsymbol{X}\left(v(t)\right) dt \tag{5}$$

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 **??**

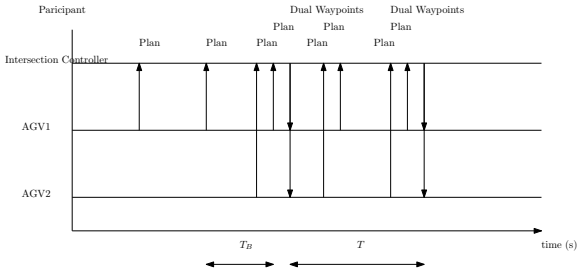

Fig. 2 Sequence diagram for two AGVs communicating with the Intersection Controller which sends a DualWaypoint message to every known AGV every T seconds, considering the latest ApproachPlans it has recieved to date.

2.5 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 [2]. In the AGV space it is common to simplify car-following with mutual exclusion of discretized roadmap segments [9] so we follow this scheme for the main results. Some results with mutual exclusion turned off are given in in Section **??** before the main results with mutual exclusion in Section **??**. 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 [13]. 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 6.

$$\alpha_{AB} = \frac{(s_B - \hat{s}) - \hat{u}(t_B - \hat{t})}{0.5(t_B - \hat{t})^2} \tag{6}$$

Within the conflict zone $s_B \leq s_A < s_C$ the acceleration α_{BC} is given by Equation 7.

$$\alpha_{BC} = \frac{(s_C - \hat{s}) - \hat{u}(t_C - \hat{t})}{0.5(t_C - \hat{t})^2} \tag{7}$$

2.6 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 8. The conflicted segment of each vehicle's path lies between s_B the smallest value of s which satisfies Equation 8 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 8. 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.

$$||X_i(s) - X_j(t)|| < W_v \quad \forall j \in N, \quad j > i \tag{8}$$

2.7 Objective

The objective to minimize the total travel time is given by Equation 9. 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 $\boldsymbol{d}_i = [d_{AB}, d_{BC}]$.

$$\begin{gathered} \min_{\boldsymbol{\phi}} \boldsymbol{J}_T = \boldsymbol{d}^T \boldsymbol{\phi} \\ \text{subject to} \\ \boldsymbol{\phi} > \phi_{min} \\ \boldsymbol{\phi}^T \boldsymbol{H_{ij}} \boldsymbol{\phi} > 0 \quad \forall i, j \in [1, p] \quad \text{with } j > i \end{gathered} \tag{9}$$

The condition $j > i$ in Equation 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.

2.8 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 $v_{max} = \sqrt{2 a_{max} d_{min}}$. Using the parameters from Table 3, the acceptable distance is d_{min} =5m.

2.9 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 onboard 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.

2.10 Conflict-zone Collision Avoidance Constraints

By definition, each intersection controller is responsible for one conflict zone, constructed as explained in Section 2.6. 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_min 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 10. Considering average speeds, the departure time ω_{max} is also linear, this is given by Equation 12.

$$\omega_i^{min} = d_{AB}\phi_{AB} = \boldsymbol{e}^T\boldsymbol{\phi}_i \tag{10}$$

Where

$$\boldsymbol{e}^T = [d_{AB}, 0] \tag{11}$$

and

$$\omega_i^{max} = [d_{AB}, d_{BC}] \begin{bmatrix} \phi_{AB} \\ \phi_{BC} \end{bmatrix} = \boldsymbol{f}^T\boldsymbol{\phi}_i \tag{12}$$

Where

$$\boldsymbol{f}^T = \begin{cases} [d_{AB}, d_{BC}], & \text{if } d_{AB} > 0 \\ [0, d_{BC}], & \text{otherwise} \end{cases} \tag{13}$$

Following [10], 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 14 are independent of the order in which AGV i and AGV j arrive.

$$|\alpha_i - \alpha_j| > \beta_i + \beta_j \tag{14}$$

Here

$$\alpha_i = \omega_i^{max} + \omega_i^{min} \tag{15}$$

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

$$\beta_i = \omega_i^{max} - \omega_i^{min} \tag{16}$$

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

$$\alpha_i = \boldsymbol{f}^T\boldsymbol{\phi}_i + \boldsymbol{e}^T\boldsymbol{\phi}_i = \mathbf{1}_i^T\boldsymbol{A}\boldsymbol{\phi}_i \tag{17}$$

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

$$\beta_i = \boldsymbol{f}^T\boldsymbol{\phi}_i - \boldsymbol{e}^T\boldsymbol{\phi}_i = \mathbf{1}_i^T\boldsymbol{B}\boldsymbol{\phi}_i \tag{18}$$

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

2.11 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 19

$$s_i > s_{i+1} + L \quad \forall i \in l \tag{19}$$

This can be expressed in terms of minimum time to collision of $TTC_{min} = 2L/(v_i + v_{i+1})$ as in Equation 20.

$$(s_i - s_{i+1})/(v_i - v_{i+1})) > TTC_{min} \tag{20}$$

It is a little awkward to capture this constraint exactly using the average speed on two segments. This approximation in Equation 21 was tested.

$$(s_i - s_{i+1})(\phi_i - \phi_{i+1})) > TTC_{min} \tag{21}$$

2.12 Non-Convex Quadratic Constraints Optimal Intersection Control

Equation 14 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 22.

$$[\boldsymbol{\phi}_i^T, \boldsymbol{\phi}_j^T] \begin{bmatrix} \boldsymbol{\Lambda}_{ij}^{ii} & \boldsymbol{\Lambda}_{ij}^{ij} \\ \boldsymbol{\Lambda}_{ij}^{ji} & \boldsymbol{\Lambda}_{ij}^{jj} \end{bmatrix} > 0 \tag{22}$$

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

$$\boldsymbol{\Lambda}_{ij}^{ii} = (\boldsymbol{A}_i - \boldsymbol{B}_i)\mathbf{1}_i\mathbf{1}_i^T(\boldsymbol{A}_i + \boldsymbol{B}_i) \tag{23}$$

$$\boldsymbol{\Lambda}_{ij}^{jj} = -(\boldsymbol{A}_j + \boldsymbol{B}_j)\mathbf{1}_j\mathbf{1}_j^T(\boldsymbol{A}_j + \boldsymbol{B}_j) \tag{24}$$

$$\boldsymbol{\Lambda}_{ij}^{ij} = \boldsymbol{\Lambda}_{ij}^{jiT} = -(\boldsymbol{A}_j + \boldsymbol{B}_j)\mathbf{1}_j\mathbf{1}_i^T(\boldsymbol{A}_i + \boldsymbol{B}_i) \tag{25}$$

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 $j < i$.

2.13 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 14 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

$$\omega_i^{max} > \omega_{i+1}^{min} \tag{26}$$

This can be expressed as

$$\boldsymbol{e}_i^T\boldsymbol{\phi}_i > \boldsymbol{f}_{i+1}^T\boldsymbol{\phi}_{i+1} \tag{27}$$

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

$$\boldsymbol{Q}^{ij}\boldsymbol{\phi} = \begin{bmatrix} 0 & \dots & & \\ \dots & \boldsymbol{e}_i^T & -\boldsymbol{f}_{i+1}^T & \dots \\ & & \dots & 0 \end{bmatrix} \begin{bmatrix} \vdots \\ \boldsymbol{\phi}_i \\ \boldsymbol{\phi}_{i+1} \\ \vdots \end{bmatrix} \tag{28}$$

The pairwise Q^{ij} matrices are added together to get A_{ub} in Equation 29,

$$A_{ub}\boldsymbol{\phi} > 0 \tag{29}$$

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

$$\boldsymbol{p}_k = [1/v_{AB}, 1/v_{BC}] \tag{30}$$

are included in Equation 31

$$\boldsymbol{e}^T\boldsymbol{\phi}_i > \boldsymbol{f}^T\boldsymbol{p}_k \tag{31}$$

here $\boldsymbol{f}$ defined in Equation 13.

2.14 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.

3 Numerical 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 3. 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 3.

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

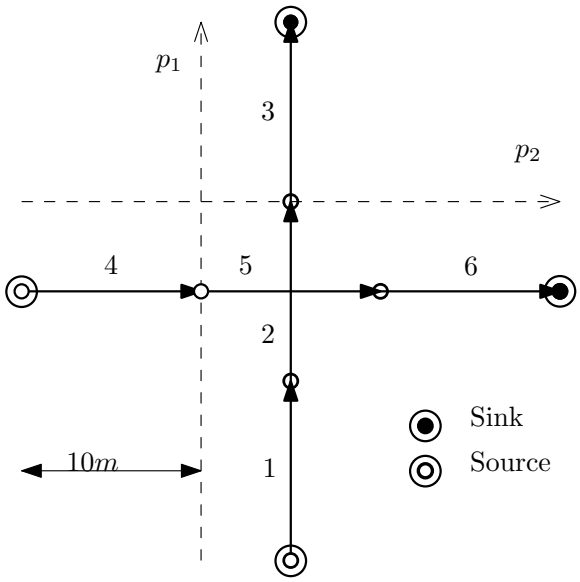

Fig. 3 Intersection layout with two conflicting routes.

	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 2 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 3 Intersection performance over 30 crossings with three different controllers for the HLHT scenario.

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 × markers for vehicles travelling along the x axis and △ 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 departure time and adjusting speeds in advance is clear from comparing the optimal controllers in Figure 3 and Figure 3 with the Semaphore approach in Figure 3. This corresponds to a reduction in delay of 4.85 seconds per vehicle according to Table 3.

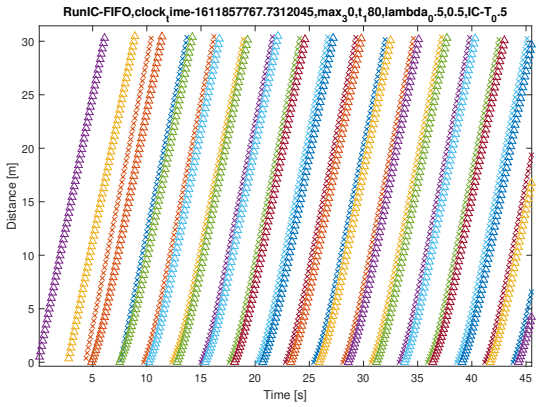

Fig. 4 Position-Time trace for HLHT Scenario under FIFO controller

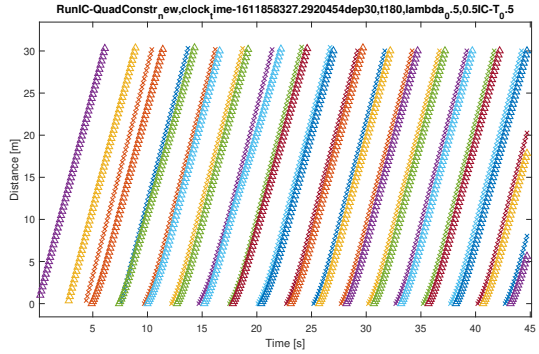

Fig. 5 Position-Time trace for HLHT Scenario under Quadratic Constraints controller

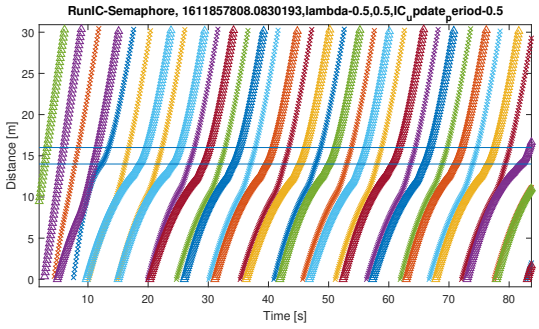

Fig. 6 Position-Time trace for HLHT Scenario under Semaphore controller

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

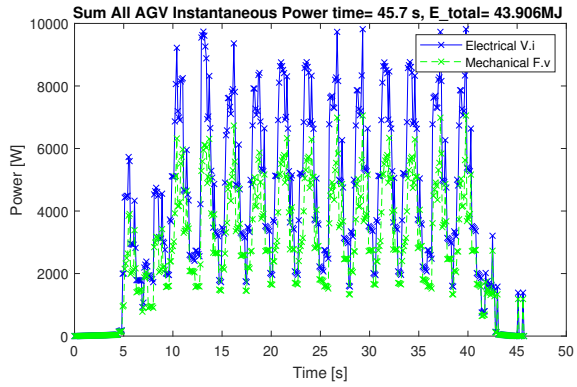

Fig. 7 Power Dissipation-Time trace for HLHT Scenario under FIFO controller

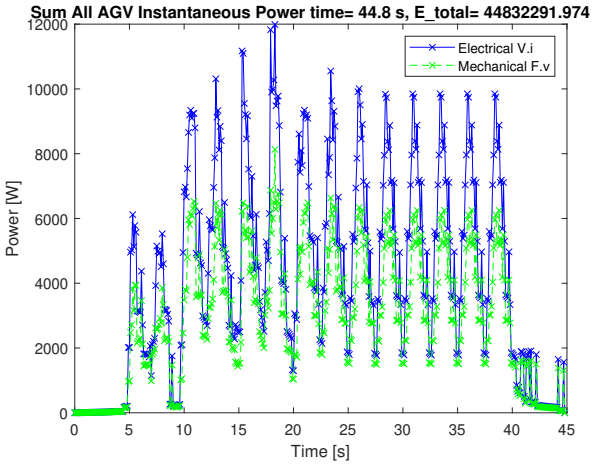

Fig. 8 Power Dissipation-Time trace for HLHT Scenario under Quadratic Constraints controller

vary the departure order. However, the departure order in Figure 3 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 3, 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

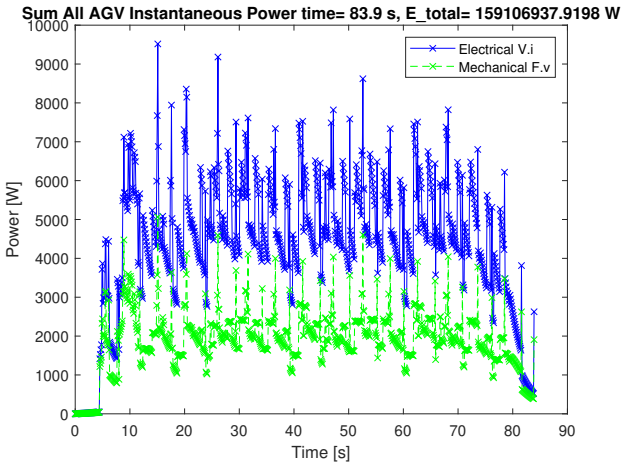

Fig. 9 Power Dissipation-Time trace for HLHT Scenario under Semaphore controller

MJ total energy compared to 44.8MJ. A spike in usage at around 18 seconds can be seen in Figure 3, possibly this corresponds to a change in order which leads to lower delay but uses some extra energy.

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

The optimization problem with quadratic constraints described in Section 2.12 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 [1]. Trust region methods make use of the exact 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[6]. 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 [27]. 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 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.

References

1. (2019) Optimization and Root Finding (scipy.optimize) minimize(method='trust-constr'). URL https://docs.scipy.org/doc/scipy/reference/optimize.minimize-trustconstr.html/hash/optimize-minimize-trustconstr
2. Baz A, Yi P, Qurashi A (2020) Intersection Control and Delay Optimization for Autonomous Vehicles Flows Only as Well as Mixed Flows with Ordinary Vehicles. Vehicles 2(3):523–541, DOI 10.3390/vehicles2030029
3. Busquets D, Francis R, Tsotskac C, North R, Galatioto F, Franco P, Guichard R, Tusting R, McCormick E, Ravenscroft G, Fletcher G, Jenkins E (2016) Distributed Simulation Using SUMO. In: SUMO 2016 – Traffic, Mobility, and Logistics Proceedings, Deutsches Zentrum für Luft- und Raumfahrt e. V., Berlin-Aldershof, pp 51–60, URL https://elib.dlr.de/106342/1/SUMOconference_proceedings_2016
4. Cardarelli E, Digani V, Sabattini L, Secchi C, Fantuzzi C (2017) Cooperative cloud robotics architecture for the coordination of multi-AGV systems in industrial warehouses. Mechatronics 45:1–13, DOI 10.1016/j.mechatronics.2017.04.005
5. Chen L, Englund C (2016) Cooperative Intersection Management: A Survey. IEEE Transactions on Intelligent Transportation Systems 17(2):570–586, DOI 10.1109/TITS.2015.2471812
6. Conn AR, Gould NIM, Toint PL (2000) Trust region methods. SIAM
7. Dai P, Liu K, Zhuge Q, Sha EH, Lee VCS, Son SH (2017) A Convex Optimization Based Autonomous Intersection Control Strategy in Vehicular Cyber-Physical Systems. Proceedings - 13th IEEE International Conference on Ubiquitous Intelligence and Computing, 13th IEEE International Conference on Advanced and Trusted Computing, 16th IEEE International Conference on Scalable Computing and Communications, IEEE Internationa pp 203–210, DOI 10.1109/UIC-ATC-ScalCom-CBDCom-IoP-SmartWorld.2016.0050
8. Digani V, Caramaschi F, Sabattini L, Secchi C, Fantuzzi C (2014) Obstacle avoidance for industrial AGVs. In: Proceedings - 2014 IEEE 10th International Conference on Intelligent Computer Communication and Processing, ICCP 2014, IEEE, pp 227–232, DOI 10.1109/ICCP.2014.6937001
9. Digani V, Sabattini L, Secchi C, Fantuzzi C (2014) Hierarchical traffic control for partially decentralized coordination of multi AGV systems in industrial environments. Proceedings - IEEE International Conference on Robotics and Automation pp 6144–6149, DOI 10.1109/ICRA.2014.6907764
10. Digani V, Hsieh MA, Sabattini L, Secchi C (2019) Coordination of multiple AGVs: a quadratic optimization method. Autonomous Robots 43(3):539–555, DOI 10.1007/s10514-018-9730-9, URL https://doi.org/10.1007/s10514-018-9730-9
11. Dresner K (2008) A Multiagent Approach to Autonomous Intersection Management. Journal of Artificial Intelligence Research 31:591–656
12. Erős E, Dahl M, Bengtsson K, Hanna A, Falkman P (2019) A ROS2 based communication architecture for control in collaborative and intelligent automation systems. Procedia Manufacturing 38:349–357, URL http://arxiv.org/abs/1905.09654, 1905.09654

13. He Y, Jia Z, Cheng Y, Li Z, Wang L, Fu J (2020) Modeling and Simulation of Heterogeneous Traffic Flow in the Vicinity of Intersections Considering Communication Delay. Chinese Control Conference, CCC 2020-July:5665–5670, DOI 10.23919/CCC50068.2020.9189519
14. Horni A, Nagel K (2016) Chapter 4: More About Configuring MATSim. In: Horni A, Nagel K, Axhausen KW (eds) The Multi-Agent Transport Simulation MATSim, Ubiquity Press, London, chap Chapter 4:, pp 35–50, DOI 10.5334/baw, URL http://dx.doi.org/10.5334/baw.4
15. Kara D, Koesling S, Kretz T, Laugel Y, Reutenauer F, Schubert F (2014) Microsimulation – a robust technical planning method with strong visual output. Proceedings of the Institution of Civil Engineers - Civil Engineering 167(5):17–24, DOI 10.1680/cien.13.00015
16. Lambert E (2020) CustomFleetSim. URL https://github.com/tsedl/custom-fleet-sim.git
17. Lambert E, Romano R, Watling D (????) Optimal smooth paths based on clothoids for car-like vehicles in the presence of obstacles. International Journal of Control, Automation and Systems
18. Lambert ED, Romano R, Watling D (2020) Intersection Platooning for Distributed Conflict Resolution of an AGV Fleet. In: IEEE International Conference on Automation Science and Engineering, TBA, Hong Kong, pp 1–4
19. Lambert ED, Romano R, Watling D (2020) Simulating Decentralized Platooning for Coordinated Conflict-Free Motion of Mobile Robot Fleets. ACM International Conference Proceeding Series pp 11–15, DOI 10.1145/3402597.3402603
20. Lenorzer A, Casas J, Dinesh R, Zubair M, Sharma N, Dixit V, Torday A, Brackstone M (2015) Modelling and simulation of mixed traffic. In: Australasian Transport Research Forum (ATRF), 37th
21. Liu R (2005) The DRACULA Dynamic Network Microsimulation Model. In: Kitamura R, Kuhawara M (eds) SIMULATION APPROACHES IN TRANSPORTATION ANALYSIS: Recent Advances and Challenges, xiii 399 p edn, Springer, chap 2. The DRA, pp 23–56, URL http://www.springer.com/978-0-387-24108-1
22. Niemueller T, Karpas E, Vaquero T, Timmons E (2017) Planning and Execution Competition for Logistics Robots in Simulation Rules and Regulations. In: Internaitonal Conference on Automated Planning and Scheduling (ICAPS) 2017, Pittsburgh, USA, pp 1–30
23. Rivera ZB, De Simone MC, Guida D (2019) Unmanned ground vehicle modelling in Gazebo/ROS-based environments. Machines 7(2):1–21, DOI 10.3390/machines7020042
24. Sarlioglu B (2016) Understanding Electric Motors and Loss Mechanisms. Tech. rep., Wisconsin Electric Machines and Power Electronics Consortium, URL https://www.irc.wisc.edu/export.php?ID=421
25. Vaughan R (2008) Massively multi-robot simulation in stage. Swarm Intelligence 2(2-4):189–208, DOI 10.1007/s11721-008-0014-4
26. Walenta R, Schellekens T, Ferrein A, Schiffer S (2017) A decentralised system approach for controlling AGVs with ROS. 2017 IEEE AFRICON: Science, Technology and Innovation for Africa, AFRICON 2017 pp 1436–1441, DOI 10.1109/AFRCON.2017.8095693
27. Wang AL, Kilinc-Karzan F (2019) The Generalized Trust Region Subproblem: solution complexity and convex hull results. Mathematical Programming pp 1–29, URL http://arxiv.org/abs/1907.08843, 1907.08843
28. Yan Z, Fabresse L, Laval J, Bouraqadi N (2017) Building a ROS-Based Testbed for Realistic Multi-Robot Simulation: Taking the Exploration as an Example. Robotics 6(3):21, DOI 10.3390/robotics6030021