My Thesis Title

Edward Derek Lambert

University of Leeds

Institute for Transport Studies

Submitted in accordance with the requirements for the degree of

Doctor of Philosophy

April, 2021

Intellectual Property Statement

The candidate confirms that the work submitted is his own and that appropriate credit has been given where reference has been made to the work of others.

Acknowledgements

Thanks everyone.

Abstract

C60 is pretty awesome for many reasons.

Contents

List of Figures

List of Tables

Abbreviations

AC	Alternating Current	PCAR	Point Contact Andreev Reflections
BCS	Bardeen-Cooper-Schrieffer	MR	Magnetoresistance
DC	Direct Current	FET	Field Effect Transistor
FWHM	Full Width Half Maximum	UHV	Ultra High Vacuum

Chapter 1

Robotic Path Planning Literature Review

1.1 Introduction

There has been more than 50 years of progress in robotic path planning, at least a substantial proportion regarding the paths of wheeled mobile robots moving in a two dimensional plane. The goal of this review is not to provide a complete reference, but to justify a certain approach to the problem which is appropriate for the domain of car-like automated vehicles moving goods in automated warehouses.

The central problem is that of a vehicle which follows a defined reference path through a known environment. In some situations the environment may change in such a way that the reference path is no longer feasible. In this case a new path must be generated based on the local information available to the vehicle, and that available from communicating vehicles. This is called Adaptive Local Planning. There are certain requirements for Adaptive Local Planning which are important for the automated warehouse domain more than others. The review reveals that is is more useful to focus on trajectories for the adaptive planning. The trajectories must be feasible in the sense that they meet obstacle avoidance constraints, and in the sense that they meet differential constraints arising from dynamic considerations of the vehicles motion. Furthermore, a guarantee that a path will be found if one exists is a desirable property. It is also desirable for the algorithm to generate a path in a consistent execution time so that the path is always available to the motion control system. In many cases these are conflicting goals: a planner with a hard time limit will not be able to search all possibilities exhaustively.

1.2 Planning Architectures

[2] divide architectures of automated vehicles into three broad categories: Traditional; Behaviour aware, and End-to-end machine learning. Traditional approaches are hierarchical, breaking the problem down into layers including perception, behaviour choice, motion planning, and feedback control. A variety of techniques can then be used at each layer. Perception covers the creation of a representation of the environment from various sensors and lies outside the scope of this review. In the DARPA Urban Challenge contestants this often consisted of a state machine which used expert rules to decide the best manoeuvre such as change lane to be executed by the motion planner [3]. The motion planner generated detailed action sequences - trajectories which

provide the reference for the feedback control component.

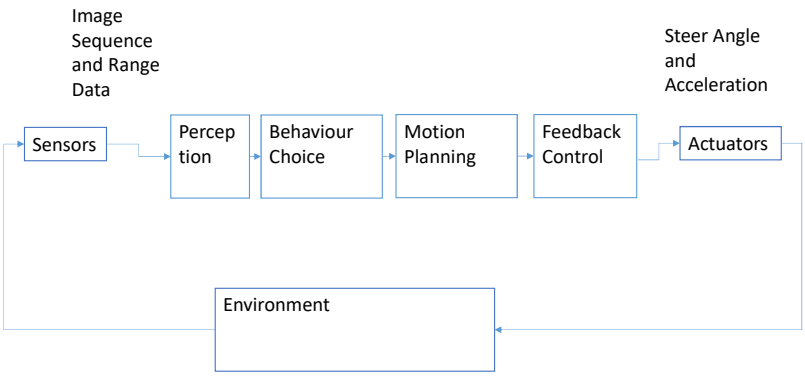


Figure 1.1: Logical flow of hierarchical planning

Outcomes can be improved by combining behaviour choice with the motion planner, evaluating different behaviour options to choose the best. For multi-vehicle situations this can lead to interesting behaviour as some model of other vehicles decisions is needed in order to evaluate possible control actions. This could be obtained by inter-vehicle communication or traffic rules based and behavioural models of human traffic participants. At this stage, bringing in multiple participants is also out of scope, but the motion planning component may be useful to inform the type of plans which it is possible to exchange by inter-vehicle communication. Understanding the detail available in a single vehicle's trajectory plan determines the information it can share and also the way it can adapt to the plans of others.

An interesting recent addition to the list of architectures, end-to-end planning 1.2 eschews traditional distinctions such as perception from planning to instead train an

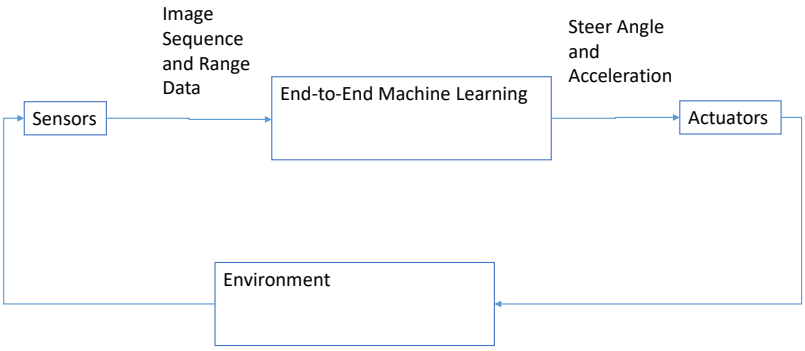


Figure 1.2: Logical flow of end-to-end planning

artificial neural network (ANN) on image data from example test drives, so they are able to produce suitable steering commands based on the current video scene in [4]. To get good results it was necessary to manipulate the training data by creating duplicate runs with a perspective offset, and a larger steering angle to correct the position. Otherwise there were not enough instances of large steering corrections in the data for a basic proportional controller to be learned which steered more vigorously the further it got from the road centre. Given the algorithm used to manipulate the data fully defined the steering behaviour it is hard to see the benefit of using machine learning in this case. It may be more appropriate to use the ANN to provide the behaviour choice and not the closed loop control, similar to [5] where the ANN outputs five states - maintain, accelerate, decelerate, turn-left, turn-right, and operates on a simplified model of the traffic detected based on a grid. The actual actuator control - steering angles and acceleration can be completed separately. This should reduce training time by reducing the complexity of the ANN (fewer layers and fewer outputs) and also reducing the amount of noise as the environment representation is quite clean, reducing the chance of over-fitting. ANN based architectures are a useful complement to traditional methods and may be particularly appropriate when predicting the future

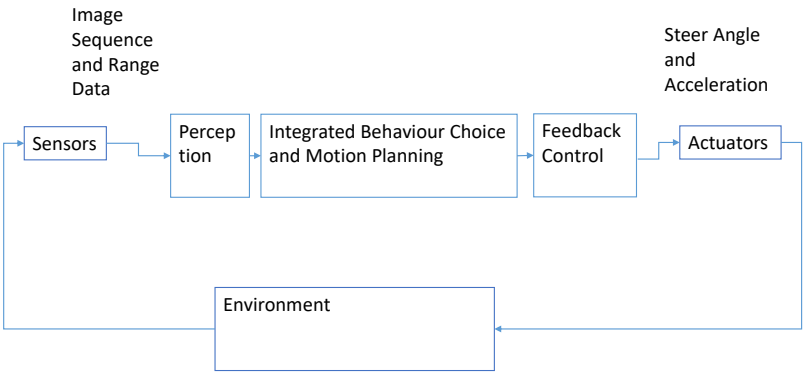


Figure 1.3: Logical flow of interactive behaviour aware planning

actions.

Interactive behaviour aware planning is useful to concentrate on because they can utilize component from traditional planning where they are best suited such as feedback control and machine learning techniques where they are most effective, such as in behaviour choice.

Hierarchical planning techniques are often based on sampling at some resolution from the configuration space, to create a discrete approximation to the problem which can be solved with graph techniques such as Dijkstra. These are able to guarantee resolution completeness, that up to the sampling resolution if a path exists it will be found.

1.3 Cut from JIRS Introduction

1.3.1 Motivation for Path Adaptation

Adaptive local path planning is important for the creation of robotic systems which can perform convincingly in variable environments. The particular environment of interest is one of bounded variation: there exists a fixed path which is preferable in most cases,

but in certain circumstances it can be invalidated, for example by an obstacle. In this case it may still be possible for the robot to complete its task with a new path avoiding the obstacle, while keeping within a predefined tolerance of the original path to manage predictability. This paper will detail one approach for creating smooth paths for an autonomous vehicle with differential constraints on its motion, and compare two methods for adapting this path in response to an obstacle detected by on-board sensors.

1.3.2 Paths and Trajectories

A path is a continuous sequence of configurations. A trajectory is a continuous sequence of configurations each valid at a particular time. A path can always be extracted from a trajectory by discarding the temporal information, but an additional procedure is required to annotate a path with timing information usually based on a choice of speed profile. To ease computation both sequences may be obtained by dividing planning into two scales.

1.3.3 Division Based on Scale

Large scale trajectory plans cover longer time periods so they can be called strategic plans by an analogy with business plans which cover longer time periods. At the other end of the spectrum small scale trajectory plans take place over shorter time periods so they might be called tactical plans.

Large scale path plans, having no timing information are usually called global plans. This is to distinguish them from local plans which only cover the immediate vicinity around the robot.

There is no widely accepted scale threshold at which a plan is considered global, as different robots have different size operating domains. As a guide, a planner would be considered global if it is able to find a path between any two valid configurations in the entire domain of operation. A planner would be described as local if there is any limitation on the distance between the start and goal locations over which it can find a path if one exists.

Global planning algorithms such as Probabilistic Roadmaps, dense random trees all require a local planner component. This component must be sufficient to connect nearby configurations (reach any final configuration) but does not have to account for obstacles. Obstacles are handled at the global layer. The local planner joins two nearby

configurations by rejecting links which intersect with obstacles.

1.3.4 Division Based on Execution

There is an alternative definition of the local planner which is used by [6], following the manual for ROS (Robot Operating System), a collection of software libraries (middleware) useful for creating robots [7]. Here a local planner is one which is able to generate a feasible path between any two adjacent nodes in the roadmap. It works in conjunction with a roadmap planner which efficiently produces large scale plans using Dijkstra or a related graph method. According to [1] this approach is known Integration Planning, where the global reasoning is provided by a roadmap planner and augmented with a local trajectory planner which accounts for obstacles. This can comprise either a System for Tactical Planning which recomputes at high frequency a path to the target location or a System of Path Deformation which alters the reference path based on sensor data.

This is different from the definition of a local planner in Section 1.3.3 where the local planner is an algorithm with limited capabilities, which is utilised by an obstacle avoiding global planner. To create a ROS *local* trajectory planner, an algorithm for global path planning of the type classified in Section 1.3.3 would be needed, along with an appropriate speed profile. This is because a core function of the ROS local trajectory planner is obstacle avoidance.

1.3.5 Problems with Shifted Curves and the PAN-Robots EU Project algorithm

One promising technique for tactical planning is generating a set of alternatives with different offsets from the reference path and choosing the one satisfying obstacle constraints as in [8]. This is a very convincing solution to the stated problem of path modification with limited variation. It is comparable to the method presented here as the form of the path is fixed but will always be suboptimal unless the number of alternatives is made extremely high.

Shifted curves inevitably compromise solution quality by limiting the search space. This is one example of an approach based on sampling from the search space. More sampling based methods are reviewed in Section 1.4.

[9] uses an algorithmic method, which divides an avoidance manoeuvre into three

stages and generates a simple cubic section for each: a lane change with an s-curve, passing the obstruction, and another lane change back onto the reference path. Collision checking is not actually addressed in detail, but described as a comparison of the free space 'size' with the size of the AGV. I am left with several questions, such as: What dimension is compared? Does it depend on the avoiding path and obstacle and AGV shape?

»¿

1.3.6 Root Finding

6. Root finding is used by [10], where a modified bisection method is used to find the roots of the lateral position error to a target final pose in (α, δ) parameters space. By matching the curvature and heading at the end of the curve, a sequence of two or four clothoids is described by only the sharpness α and deflection δ of the first segment.

1.3.7 Constructive Polylines

7. An earlier geometric method from [11] is able to find the parameters for a CC-Path based on clothoids to join any two poses with zero curvature at the start and end. This requires that each pair of clothoids be symmetrical and end with zero curvature. The authors call this the "generic turn" procedure. This approach is developed further in [12] with a one dimensional search to find the least maximum sharpness. By the definition of sharpness as the rate of change of curvature with path length, this leads to the path which can be traversed at a given speed with the slowest steering rate. This should lead to smooth and easily drivable paths.

1.3.8 Tactical Motion Planning

8. The problem addressed in the current work is that of computing an optimal alternative path, given an obstruction which invalidates the reference path. It is important in an industrial environment where reference paths can be designed and are preferable to any seemingly convenient adaptive path for reasons of safety and predictability. The only reason a new path is needed is in the case the reference is invalid, due to an unexpected obstacle. This problem will not result in a complex obstacle maze, and if it does it is undesirable for a robot to navigate it. These are industrial robots, not explorers, but if there is an obvious and safe workaround to a problem they should be able to

Given:

1. A workspace **W** is either in R^2 or R^3
2. An obstacle region **O**
3. A robot defined in **W** consisting of one or more rigid bodies
4. A configuration space **C** comprising $\mathbf{C_{obs}}$ and $\mathbf{C_{free}}$
5. An initial configuration $q_I \in \mathbf{C_{free}}$
6. A goal configuration $q_G \in \mathbf{C_{free}}$

Compute a continuous path $\tau : [0, 1] \rightarrow C_{free}$ such that $\tau(0) = \mathbf{q_I}$ and $\tau(1) = \mathbf{q_G}$

Figure 1.4: The piano mover's problem - geometric path planning.[1]

detect it and plan accordingly. If there is no clear solution, it is always better to alert operators and wait for assistance, than to forge into the unknown and risk creating unsafe and unpredictable situations.

1.4 Sampling Based Planning Techniques

1.4.1 Introduction

Geometric path planning or the piano mover's problem consists of finding a sequence of configurations between an origin and a destination within a field of obstacles. The problem is known to be PSPACE hard, that is to require at least a polynomial amount of memory to solve. It has been a topic of active research in robotics and animation for the last several decades and many practical solutions have been developed, especially for the case of a two dimensional workspace with polynomial obstacles. An overview is given in Handbook of Robotics, Part A: Robotic Foundations - Motion Planning [1].

Typically approaches make some approximation to make the problem tractable, at the cost of some aspect of global optimality of the solution. Approaches may be divided roughly into Mathematical Programming based and Sampling Methods, although there are others such as Potential Field methods and many combinations.

Sampling methods for motion planning can be divided into deterministic and ran-

domized sampling. Deterministic samplers subdivide state space into a uniform lattice such as a grid or a more exotic symmetrical structure designed to span the search space effectively. Randomized sampling methods fall into two main categories: Rapidly Exploring Trees (RRTs) [13] and Probabilistic Roadmaps (PRM). Probabilistic roadmaps are typically used for static environments, where it is advantageous to make multiple queries on the same roadmap, for a robot performing different taks within the same environment. This is because it is time consuming to produce a new roadmap but quick to make additional queries on it. RRTs are more useful when the environment is changing rapidly such as a robot exploring new areas. The tree is quick to construct but there is little advantage to making subsequent queries.

1.4.2 State Lattice Planners

Sampling from the state space in a regular lattice to create a graph for efficient searching is one of the first practical approaches to path planning for mobile robots. This is because the computational complexity was low enough for real-time systems provided certain assumptions held. Typically the lattice took the form of a regular grid, either four connected or eight connected by straight lines. Each node represented the x-y state of a holonomic robot, able to move in any workspace direction freely. The holonomicity assumption allows many interesting path finding behaviours to be studied, but typically does not hold for practical systems like cars or fork lift trucks with Ackerman steering which are subject to differential constraints.

Smoothing

One way of incorporating differential constraints into the lattice planning approach is to fit a smooth curve which respects the constraints such as a cubic spline as closely as possible to use nodes returned by the planner. The difficulty here is that the smooth curve does not exactly follow the edges in the graph which were used to calculate the route cost and check for obstacle intersection. If the cost of the smooth path is different to that of the approximation used in the lattice, any optimality guarantees provided by the planning method such as Dikstras algorithm are compromised. Worse, the smooth path must be checked against the obstacle map for collisions, and it is not clear how the path should best be modified if any collisions are detected. For an example of smoothing with clothoid curves see [14] [15].

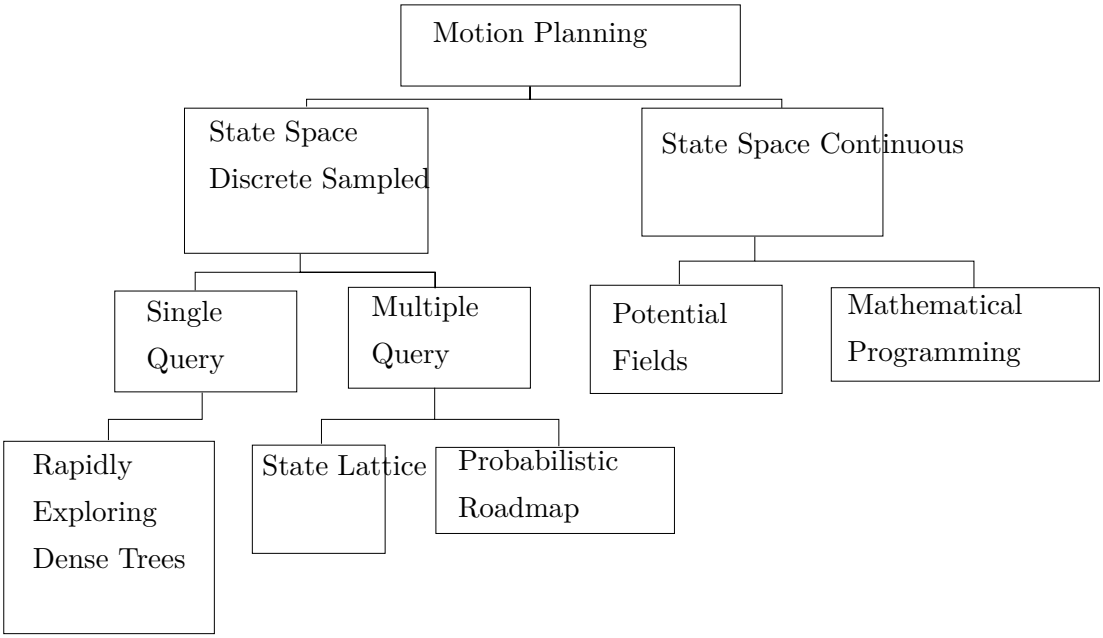


Figure 1.5: Hierarchy of planning approaches. See 'Handbook of Robotics, Part A: Robotic Foundations - Motion Planning' [1] for more details.

Differential Constraints

It is possible to carefully construct a lattice which captures differential constraints as detailed by Pivtoraiko et al [16].The trick is calculating the control inputs required to join a set of vertices which span state space (inverse kinematics) or sampling from the control space in order to generate a set of vertices (forward kinematics). Both methods have complications, for example the shape of the lattice must be known ahead of time to be reachable with a simple set of primitives for the inverse method. For the forward method the difficulty is in the choice of interval in control space that leads to complete and uniform coverage of state space. These difficulties are resolved for one example configuration in [16] with 16 discrete headings, 8 levels of curvature and a total of 192 controls. The balance to be struck in the choice of resolution is that higher resolutions lead to a higher branching factor, increasing the memory required to represent a given area, but lower resolutions will give suboptimal paths. Imagine following a straight wall which does not align with one of the 16 cardinal directions in the lattice, but halfway between. The planner with return an optimal path which alternates between the two closest cardinal directions, rather than a straight line aligned with the wall. With a back

of the envelope calculation, increasing the resolution to 10mrad to minimize artefacts increases the number of nodes required per square metre to exceed the number of atoms in the known universe. This trade off is the biggest obstacle to the use of lattice planners for general path planners and led to the development of randomized planners, which can give satisfactory uniformity and coverage of high dimensional state space without the direct link between the branching factor and the resolution when constructing a lattice.

1.4.3 Probabilistic Roadmaps

Probabilistic Roadmaps are an example of a multiple-query sampling based planner. They are probabilistic in the sense they sample randomly from **W** to build up a connectivity graph - the roadmap. They do not inherently cope with uncertainty in the obstacle field. One important component is a local planner, which is able to generate a path between two nearby configurations and test if it intersects with any obstacles. The algorithm proceeds by testing each sample by creating a local path from the nearest point on the existing tree, and only adding the new point to the graph is the path is obstacle free. Depending on the local planner it is possible to incorporate smooth paths respecting vehicle dynamic limits, but other methods are more appropriate in cases where the obstacle field is dynamic because the simplifying assumption making roadmaps so fast is that of a static environment. Another important component is the graph method used to select the shortest path such as Dijkstra which is used to find the shortest path once the roadmap is constructed. There also must be some heuristic guiding the selection of new points.

The distinction between PRMs and lattice planners is only in the mechanism by which samples are drawn from state space to construct the roadmap [1]. They are both Roadmap planners. Either a repeating pattern or lattice is overlaid or states are selected randomly within an area of interest and discarded if they are not reachable according to the obstacle map and the local planner. This is based on the idea that randiom samples of high enough density will provide unbiased and complete coverage of the space. Particularly in the limit as the number of samples tends to infinity, a PRM may be 'probabilistically complete' so that as the number of samples tends to infinity the probability of finding a solution if one exists tends to one.

The modifications to both types of Roadmap planners which deal with a changing

environment and differential constraints on motion are very similar. Differential constraints can be incorporated into the local planner as described in 1.4.2, alththough there are fewer constraints on PRM as there is no need to use kinmatics to ensure the samples form a regular lattice.

Uncertainty

Both types of roadmap planner are useful for dealing with uncertain representations of the world where the environment is represented as an occupancy grid, a grid of values covering the space, each value representing the occupancy probability of that cell. The best path can be found directly from this representation by minimising the sum of occupancy probability of every cell traversed by the path. This may be more useful than the 'hard constraints' offered by optimization methods, because although the solution will not violate the constraints, the constraints themselves are constructed by throwing away information about the uncertainty of the environment to create a binary representation where every position in space is either inside or outside polygonal obstacles [16].

1.4.4 Rapidly Exploring Random Trees

Rapidly Exploring Random Trees are an example of a single query planner introduced by LaValle and Kuffner [13]. Most simply, a tree is constructed by sampling from state space close to an existing node in the tree. A local planner is used to generate a trajectory from the existing node to the new sample avoiding obstacle regions if one exists. If an obstacle free trajectory is found, a new node is added at the sample state with an edge from the existing node. Nearby nodes in the tree should also be checked and connected with edges if possible.

Starting with the initial position and the goal position, this process is repeated until sufficient coverage of the state space is obtained. When a new sample is reachable from both trees, origin and destination must be connected and a graph method such as Dijkstra can be used to identify the vertices through the graph which make up the shortest path between them. The tree growing phase of the algorithm can be terminated at this point but it may be possible to improve the quality of the solution by adding more reachable states to the connected graph in case a better solution can be found.

Chapter 2

Dynamic Platooning for Automated Guided Vehicles (AGV)

2.1 Introduction

AGV typically have their motion restricted to a roadmap connecting pick and drop locations [17]. This reduces the search space to make conflict free routing of multiple vehicles between different origins and destinations feasible. [18] proposes a two level decomposition of the problem with the detailed roadmap at the bottom and a higher level abstraction of zones around each intersection above. The conflict avoidance is handled with local information within each zone. The routing decisions are made at high level, utilizing a traffic model where each intersection has a fixed capacity (number of vehicles).

Coordinated conflict-free motion of a number of mobile robots in order to complete a material transfer task is important in the operation of fleets of AGV (Autonomous Guided Vehicles) used in flexible manufacturing and automated warehouses [19] and [20]. A crucial sub-problem is conflict resolution between multiple AGVs, without control of task assignment or scheduling.

It is shown, in [21], that platooning provides superior throughput to the earlier reservation based systems, and that if a solution exists it is optimal, but not that a solution exists on all roadmaps. More importantly a set of conditions which must hold for a solution to exist, is not given. The consensus algorithm in [22] also shows improved throughput in concert with a scheduling approach, but does not prove convergence. An example of a resolution complete algorithm based on spatial reservation is [23]. Neither per-intersection optimal platooning nor per-vehicle consensus have been proven complete. The lack of guarantees is an important limitation of platooning methods for collision avoidance. The research gap identified is the lack of investigation into the range of motion conflict situations that can be resolved with platooning methods.

2.2 Literature Review

AGV motion coordination can be posed as a variation of the Multiple Vehicle Routing Problem with the addition of challenging spatio-temporal constraints preventing collisions between each vehicle, as well as the usual timing and capacity constraints [24]. In [25], solutions are classified into centralized, decentralized and decoupled approaches. Each approach may be either optimal or heuristic based on whether or not they find the global minimum of some objective function.

It is also possible to classify approaches based on the limitations they place on the state space of each AGV. Most practical methods incorporate both obstacle avoidance constraints and differential motion constraints into some sort of roadmap. This is often a graph with vertices at key points in the reachable state space, connected by edges representing feasible motion between them. The effect of increasing resolution on path quality (measured by reduction in the length of the shortest path) and computation time are studied in [26].

Interest in centralized methods which plan in the full state space all of the vehicles was renewed by the development of effective numerical tools for operations research able to solve large combinatorial problems to optimality. Notably [27] found trajectories for aircraft using a linear approximation to the dynamics and obstacle constraints, allowing the use of Mixed Integer Linear Programming. The contribution of [25] is to describe a new centralized, optimal method which scales well, finding a solution to the nonlinear-program for 10 vehicles in just a few seconds using an interior point method. However, size of the combined state space explodes as more vehicles are included so centralized methods are difficult to scale up to large numbers of vehicles.

Many types of decoupled methods have been developed because breaking the problem down into sub-problems is one way to reduce computation time for practical applications. Early methods of this type were based on timed petri nets [28] and agent based models [29]. Decoupled methods may sacrifice completeness (that a solution is found if one exists) in exchange for reduced average run-time. In [30], this was shown to cause practical difficulties in the spot-welding task studied. Spot welding requires close formation control of six vehicles, and the decoupled method frequently failed to find a solution. In [31], decoupled planning (incomplete) is compared with a new multi-phase heuristic, which is complete, for 50 robots on a tunnel map and 150 on an outdoor map. Decoupled planning was consistently faster in execution and produced shorter paths for lower vehicle density however it failed to find any valid paths at all for high vehicle density (25 or more robots in tunnels and 75 or more outdoors). The multi-phase heuristic, being complete, found a solution in every test case.

More recently, [32] addressed the lack of optimality in decoupled methods operating on graphs. Optimal conflict-free motion is posed as a large Integer Linear Program. Resolution complete general purpose algorithms are used to solve it for 150 robots in just over 10 seconds. The lattice/graph construction has recently been developed fur-

ther to ensure kinematic constraints are met and improve coverage of state space around obstacles [33]. In [24], the combined problem of DCFVRP (Dispatch and Conflict-Free Vehicle Routing Problem) for flexible manufacturing is formulated as an integer program and two different decoupled algorithms are presented to solve it: local search and random search. Neither of the proposed algorithms is complete but local search found more valid solutions in the 10 random examples tested, all involving three vehicles.

Decentralized control is another option to decompose large scale problems which take too long to solve centrally [34]. Although limiting the information available to each decision maker can make reasoning about collective behavior more difficult, various attempts to decentralize conflict-free routing have been made. In the field of conflict free routing for mobile robots, [35] presents a solution which generates a graph representation of the free space - effectively a roadmap - with the property of 'collision-avoidability.' This means that every node on the critical path must be at most one move away from a node that does not obstruct the critical path. The critical path is defined as the union of all the shortest paths between pick/drop locations in the roadmap. During decentralized planning, AGVs attempt to reserve 'private zones' consisting of the node on the critical path along with all adjacent collision avoidance nodes. Each AGV has an identical roadmap, plans the shortest path to its own goal and negotiates with those nearby based on a numeric priority to reserve the nodes in its own path. An AGV requests those in its path move to their collision avoidance node, and those with a lower priority will do so. Proof is given of correctness, that deadlocks are avoided, but throughput is sub-optimal with low priority vehicles frequently forced to stop and wait. An alternative decentralized solution, based on a roadmap with two levels of detail is summarized in [36]. Conflict-free routing primarily takes place at the most detailed level, based on prioritized roadmap reservation with local negotiation to guarantee correctness [9]. In [21], the speed of the approaching AGV is optimized at each intersection in a similar way to centralized intersection platooning. The result is higher throughput as time consuming negotiation is avoided in most cases.

Congestion effects are represented by link performance functions in the approximate level graph. Intersections have a generalized cost which increases exponentially up to a fixed capacity which identified by parameter tuning, An optimal task scheduling approach based on the Hungarian Algorithm is used to solve the full DCFVRP in [37] and [36]. Traffic delays are a type of emergent behaviour and modelling is challen-

ging even in a completely automated system. This is the contribution of [38] which introduces the PRT (Probabilistic Reservation Table) to summarize the plans of robots including uncertainty so it can be in task scheduling. This approach is compared with a reservation based centralized planner as a baseline, in a simulation with 5 robots using a low level motion controller from ROS (Robot Operating System) which often fails at when two robots plan interfering paths. Congestion aware planning leads to fewer failures of the low level controller than independent planning but more than centralized planning. A more convincing comparison would be with a congestion aware planner using a deterministic congestion model instead of the PRT, but this is not reported.

Intersection control, based on platooning, is a concept developed for the operation of anticipated CARVs (Connected and Autonomous Road Vehicles). A recent review of approaches for intersection and merging coordination is given in [39]. Centralized optimization approaches improve on early ideas like First-Come-First-Served spatial reservation from [40] by minimizing fuel consumption, but the rapid increase in state space with larger numbers of vehicles will need to be addressed before large scale adoption. The communication channel connecting every vehicle with the central controller introduces a single point of failure, the reliability effect of which is difficult to evaluate in existing simulations. Moreover there are few CARVs available making a practical experiment unfeasible in most cases. Attempting to address these limitations are decentralized methods such as fuzzy-logic, virtual vehicle platooning and invariant set approaches. Notably the conditions for solutions to exist which minimize energy consumption are given in [41].

Recently [22] described an approach to the DCFVRP for flexible manufacturing based on dynamic platooning with vehicle-to-everything messaging and consensus speed control, resulting in a decentralized heuristic solution with some additional rules to ensure correct behavior and avoid deadlocks by adding a reservation protocol for some parts of the roadmap. This was combined with feedback from the queue length at different workstations in a traffic management heuristic. Simulation results show an impressive improvement compared to the first-come-first-served scheduling approach meant to represent industry standard practice.

A promising approach for intersection control applied to AGV is given in [21]. As the paths are not modified, only the speed adjusted deadlocks are proved impossible. However, a backup system based on negotiation is still required because the problem is

non-convex suggesting a feasible solution may not be found in time for certain roadmap and traffic combinations. The consensus based platooning method for local collision avoidance used in [22] is unusual in the AGV domain. That work makes no claims about completeness, but does consider the trivial consensus where all vehicles stop in a deadlock. In [42], a recent system for conflict avoidance based on time headway is shown to significantly reduce intersection crossing time and allow more vehicles to operate in the same floor-space compared to a traditional reservation based strategy. Of these different approaches to platooning for AGV coordination, the quadratic constraints of [21] are the closest to acheiving the certainty the would be required to build a distributed coordination scheme.

2.3 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 [43]. 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 [21]. The surrounding lanes are first discretized into segments. The intersection shown in Figure 2.1 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 and the other takes route 2. If they both travel at maximum speed they will collide in

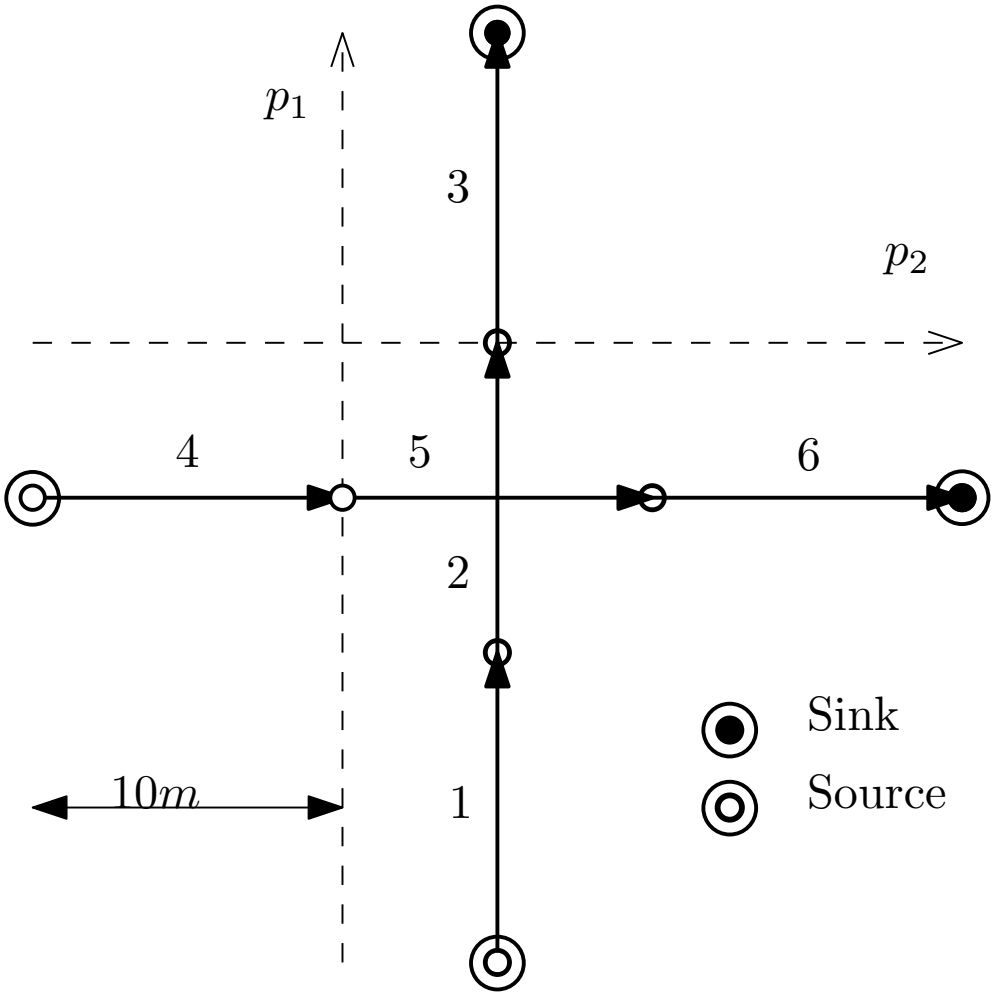


Figure 2.1: Intersection layout with two conflicting routes.

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 2.2. 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 2.9.

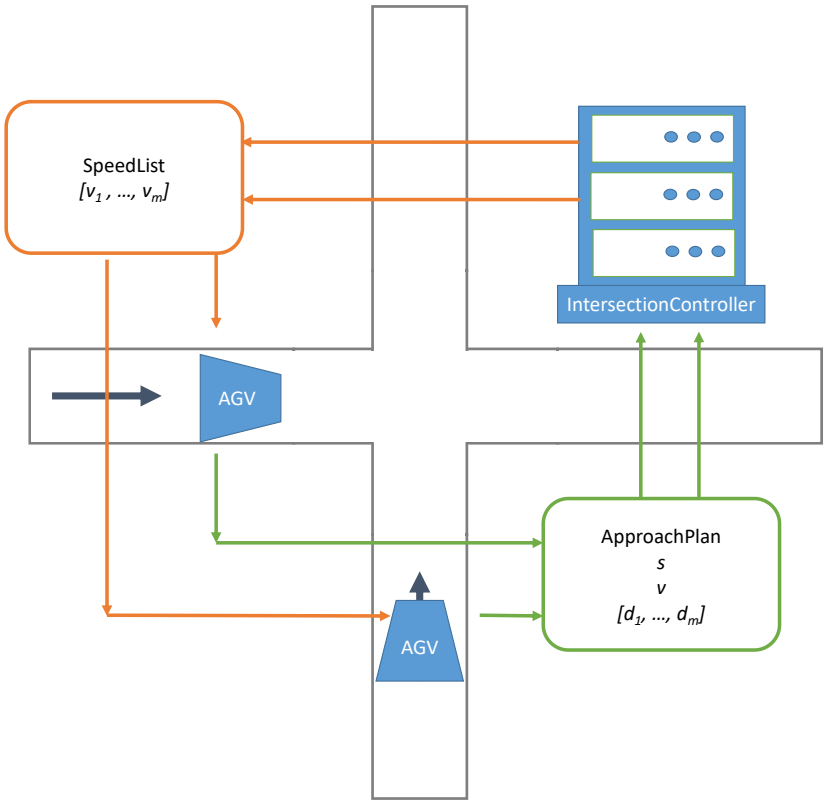


Figure 2.2: Messages exchanged by participants approaching intersection.

2.3.1 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 [44] 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 2.3 which is well known, for example see [45].

This simple circuit leads to the relationship between current and internal resistance given by Equation 2.1.

$$V_{cc} = E_D + I_a R_a \tag{2.1}$$

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

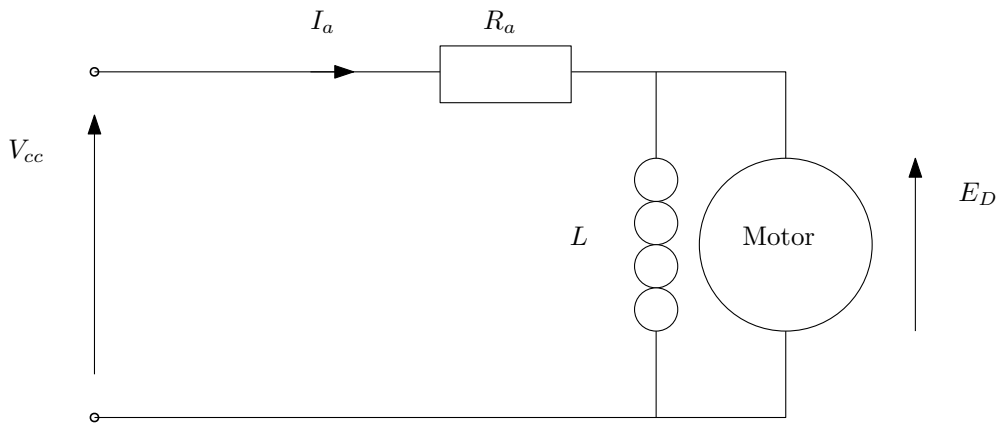


Figure 2.3: Steady state equivalent circuit for a DC motor.

such as the induction motor due to its varying excitation [46]. 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.

$$\tau = k_T I_a \tag{2.2}$$

$$E_D = \frac{rpm}{k_e} \tag{2.3}$$

There are numerous loss sources in an electric motor such as winding resistance, flux leakage, eddy currents in the core and so on [46]. 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 2.1.

2.3.2 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 2.1: 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 [47]. 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.

$$F_D = C_{drag}\rho v^2 A \tag{2.4}$$

2.3.3 Arrival Distribution

Previous work in road traffic modelling has used a variety of point distributions to model arrivals. A good summary is given in [48] or the chapter on Microsimulation in [49].

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 2.4. 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 2.7.

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

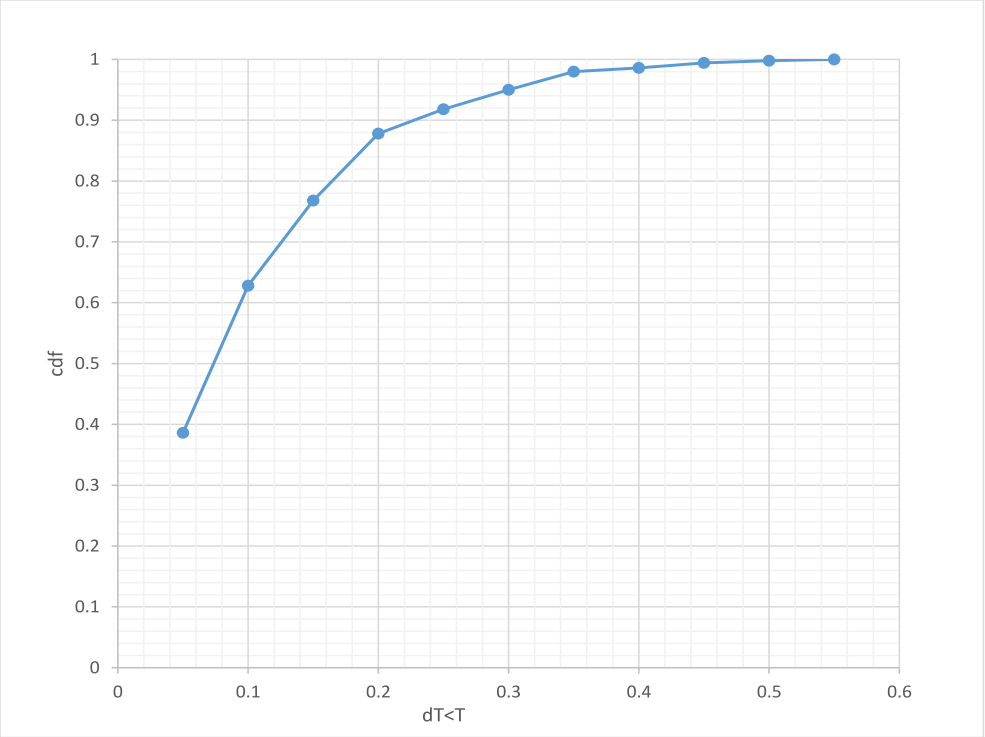


Figure 2.4: Cumulative Distribution Function of arrivals following a Poisson distribution.

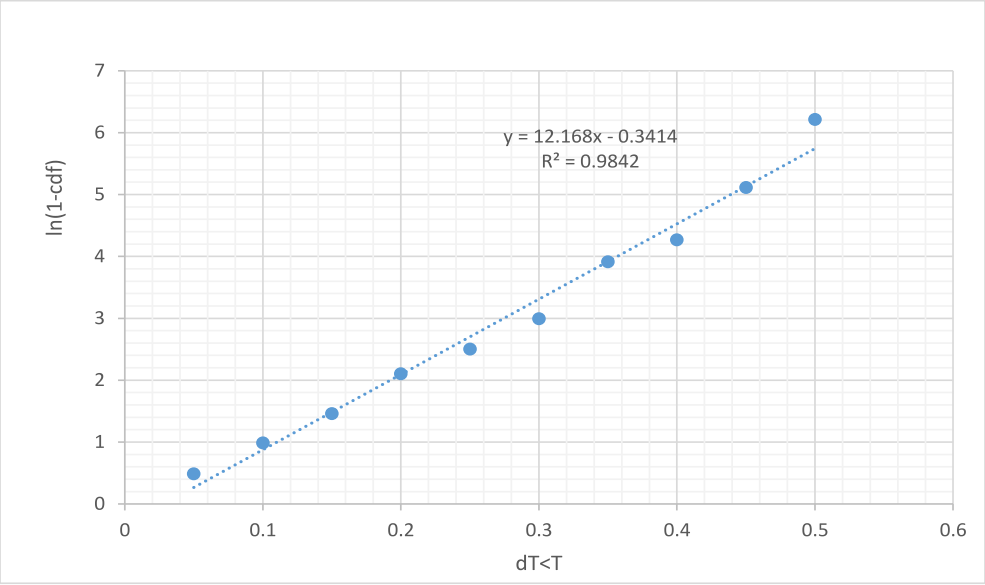


Figure 2.5: Log Cumulative Distribution Function of arrivals with linear fit.

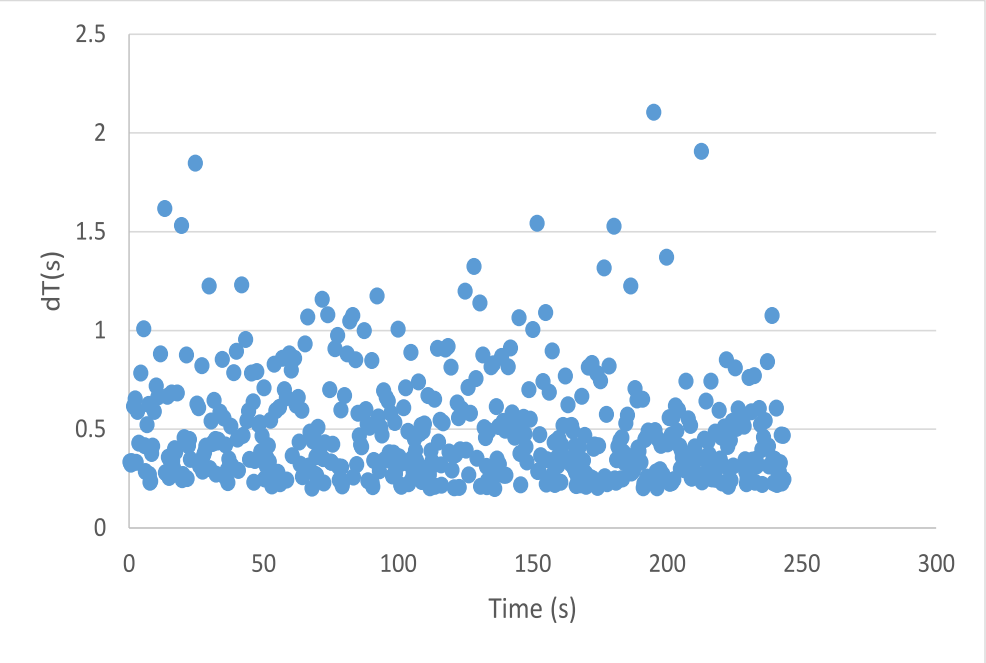


Figure 2.6: Scatter plot of 500 arrival time deltas drawn from a shifted exponential distribution with a minimum headway of 0.2 seconds.

distribution. A scatter plot of times from this distribution is shown in Figure 2.6. 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

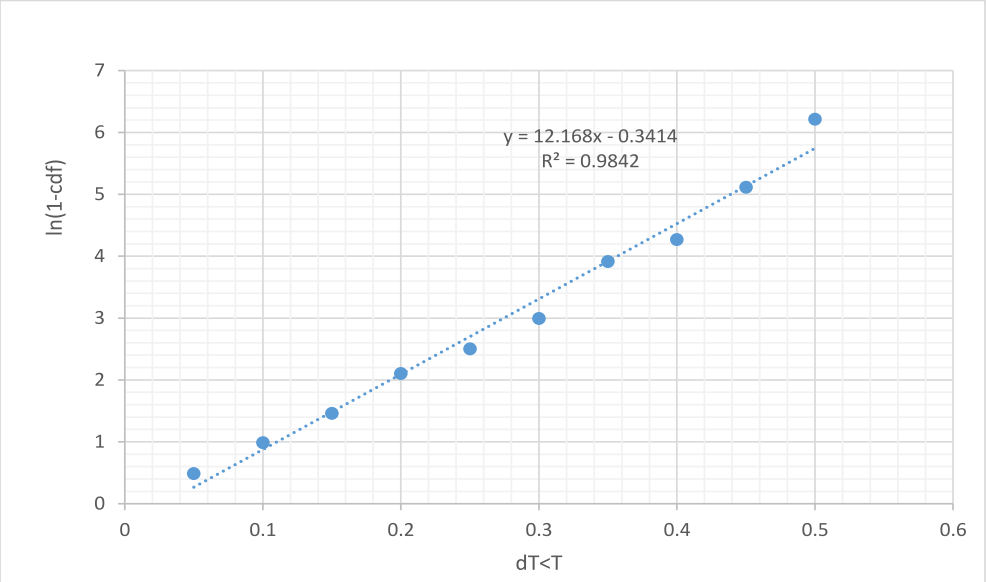


Figure 2.7: Log Cumulative Distribution Function of arrivals with linear fit.

- 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.

2.3.4 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 [21] where the intersection provides average speeds for discrete segments, and the one used by Levin and Rey [50] 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 [51].

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 [52].

[50] and [21] 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.

2.4 Methods For Zone-Based Intersection

2.4.1 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 2.10 and Equation 2.12 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 2.5

$$\boldsymbol{\phi}_i = [\phi_1, ..., \phi_{(m_i+n_i)}]^T \tag{2.5}$$

The parameters for p vehicles each traversing $(m_i + n_i)$ segments are assembled into a vector as in Equation 2.6

$$\boldsymbol{\phi} = [\boldsymbol{\phi}_\mathbf{1}^T, ..., \boldsymbol{\phi}_p^T]^T \tag{2.6}$$

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

$$\boldsymbol{d}_i = [d_1, ..., d_{(m_i+n_i)}]^T \tag{2.7}$$

and collected for p vehicles into a vector as in Equation 2.8.

$$\boldsymbol{d} = [\boldsymbol{d}_\mathbf{1}^T, ..., \boldsymbol{d}_p^T]^T \tag{2.8}$$

This leads to the minimum travel time objective in Equation 2.9.

$$\begin{gathered} \min_{\boldsymbol{\phi}} \boldsymbol{J_T} = \boldsymbol{d}^T\boldsymbol{\phi} \\ \text{subject to} \\ \boldsymbol{\phi}_{max} > \boldsymbol{\phi} > \boldsymbol{\phi}_{min} \\ \boldsymbol{\phi}^T \boldsymbol{H}_{i,j} \boldsymbol{\phi} > \mathbf{0} \quad \forall i, j \in [1, p] \quad \text{with } j > i \end{gathered} \tag{2.9}$$

The condition $j > i$ in Equation 2.9 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.

2.4.2 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 2.10. Considering average speeds, the departure time ω_i^{max} is also linear, this is given by Equation 2.12.

$$\omega_i^{min} = \sum_{k=1}^{m_i} \boldsymbol{d}_i[k]\boldsymbol{\phi}_i[k] = \boldsymbol{e}^T\boldsymbol{\phi}_i \tag{2.10}$$

where

$$\boldsymbol{e}[k] = \begin{cases} \boldsymbol{d_i}[k] & \forall k < m_i \\ 0 & \text{otherwise} \end{cases} \tag{2.11}$$

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

$$\omega_i^{max} = \omega_i^{min} + \sum_{i=1}^{n_i} \boldsymbol{d}_i[k]\boldsymbol{\phi}_i[k] = \boldsymbol{f}^T\boldsymbol{\phi}_i \tag{2.12}$$

where

$$\boldsymbol{f}[k] = \begin{cases} \boldsymbol{d_i}[k] & \forall k < m_i + n_i \\ 0 & \text{otherwise} \end{cases} \tag{2.13}$$

and n_i is the number of segments on the path of vehicle i which are conflicted. Note that Equations 2.10 and 2.12 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 2.14.

$$-t_i + t_{i+1} \leq \Delta t_i \tag{2.14}$$

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 2.15. This is

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

$$\begin{bmatrix} -d_1 & d_2 & \mathbf{0} \\ \vdots & \ddots & \vdots \\ \mathbf{0} & -d_{p-1} & d_p \end{bmatrix} \begin{bmatrix} \phi_1 \\ \vdots \\ \phi_p \end{bmatrix} \leq \begin{bmatrix} \Delta t_1 \\ \vdots \\ \Delta t_{p-1} \end{bmatrix} \tag{2.15}$$

The timing constraint between each pair of vehicles can be expressed with a modulus operator as in Equation2.16.

$$|\alpha_i - \alpha_j| > \beta_i + \beta_j \tag{2.16}$$

Here

$$\alpha_i = \omega_i^{max} + \omega_i^{min} \tag{2.17}$$

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

$$\beta_i = \omega_i^{max} - \omega_i^{min} \tag{2.18}$$

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

In matrix form

$$\alpha_i = \boldsymbol{f}^T \boldsymbol{\phi}_i + \boldsymbol{e}^T \boldsymbol{\phi}_i = \mathbf{1}_i^T \boldsymbol{A} \phi_i \tag{2.19}$$

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

$$\beta_i = \boldsymbol{f}^T \boldsymbol{\phi}_i - \boldsymbol{e}^T \boldsymbol{\phi}_i = \mathbf{1}_i^T \boldsymbol{B} \phi_i \tag{2.20}$$

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 2.16, without forcing any particular arrival order is to square both sides as to give the expression in Equation 2.21.

$$\alpha_i^2 - \alpha_j^2 - 2\alpha_i\alpha_j - (\beta_i^2 + \beta_j^2 + 2\beta_i\beta_j) > 0 \tag{2.21}$$

Collecting terms by subscript gives

$$(\alpha_i^2 - \beta_i^2) - (\alpha_j^2 + \beta_j^2) - 2(\alpha_i\alpha_j + \beta_i\beta_j) > 0 \tag{2.22}$$

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 2.23. 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.

$$\mathbf{\Lambda}_{ij} = \begin{bmatrix} \mathbf{\Lambda}_{ij}^{ii} & \mathbf{\Lambda}_{ij}^{ij} \\ \mathbf{\Lambda}_{ij}^{ji} & \mathbf{\Lambda}_{ij}^{jj} \end{bmatrix} \tag{2.23}$$

Expanding

$$\begin{aligned} \begin{bmatrix} \boldsymbol{\phi}_i^T, \boldsymbol{\phi}_j^T \end{bmatrix} \begin{bmatrix} \mathbf{\Lambda}_{ij}^{ii} & \mathbf{\Lambda}_{ij}^{ij} \\ \mathbf{\Lambda}_{ij}^{ji} & \mathbf{\Lambda}_{ij}^{jj} \end{bmatrix} \begin{bmatrix} \boldsymbol{\phi}_i \\ \boldsymbol{\phi}_j \end{bmatrix} \\ = \boldsymbol{\phi}_i^T \mathbf{\Lambda}_{ij}^{ii} \boldsymbol{\phi}_i + \boldsymbol{\phi}_j^T \mathbf{\Lambda}_{ij}^{jj} \boldsymbol{\phi}_j + \boldsymbol{\phi}_i^T \mathbf{\Lambda}_{ij}^{ij} \boldsymbol{\phi}_j + \boldsymbol{\phi}_j^T \mathbf{\Lambda}_{ij}^{ji} \boldsymbol{\phi}_i \end{aligned} \tag{2.24}$$

makes it possible to compare terms with the scalar expression in Equation 2.22. 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 $\beta_i = \mathbf{1}_T \boldsymbol{B}_i \boldsymbol{\phi}_i$

$$\mathbf{\Lambda}_{ij}^{ii} = (\boldsymbol{A}_i - \boldsymbol{B}_i) \mathbf{1}_i \mathbf{1}_i^T (\boldsymbol{A}_i - \boldsymbol{B}_i) \tag{2.25}$$

$$\mathbf{\Lambda}_{ij}^{jj} = -(\boldsymbol{A}_j + \boldsymbol{B}_j) \mathbf{1}_j \mathbf{1}_j^T (\boldsymbol{A}_J + \boldsymbol{B}_j) \tag{2.26}$$

$$\mathbf{\Lambda}_{ij}^{ij} + \mathbf{\Lambda}_{ij}^{ijT} = -2(\boldsymbol{A}_j + \boldsymbol{B}_j) \mathbf{1}_j \mathbf{1}_i^T (\boldsymbol{A}_i + \boldsymbol{B}_i) \tag{2.27}$$

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 [53]. 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 2.16, 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 2.28. 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,

$$\begin{aligned} \alpha_i - \alpha_j &> \beta_i + \beta_j \quad && \text{if } \alpha_i > \alpha_j \\ \alpha_i - \alpha_j &< -(\beta_i + \beta_j) && \text{otherwise} \\ \text{where } \alpha_i, \alpha_j, \beta_i, \beta_j &> 0 \end{aligned} \tag{2.28}$$

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 2.29.

$$\begin{aligned} \alpha_i - \alpha_j + b_{i,j}M &> \beta_i + \beta_j \\ \alpha_i - \alpha_j - (1 - b_{i,j})M &< -(\beta_i + \beta_j) \\ \text{where } b_{i,j} &\in [0, 1] \\ M >> \alpha_i, \alpha_j, \beta_i, \beta_j & \end{aligned} \tag{2.29}$$

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 [54]. 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 2.17 and 2.18 with a constant $p_i = g_i\phi_{min}$.

$$\alpha_i = 2\omega_i^{min} + p_i \tag{2.30}$$

$$\beta_i = p_i \tag{2.31}$$

where g_i is the length of the conflicted segments in the plan of vehicle i. In this case Equation 2.29 can be restated

$$\begin{aligned} \omega_i^{min} - \omega_j^{min} + b_{i,j}M &> p_j \\ \omega_i^{min} - \omega_j^{min} - (1 - b_{i,j})M &< -p_i \\ \text{where } b_{i,j} &\in [0, 1] \end{aligned} \tag{2.32}$$

2.5 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 [50].

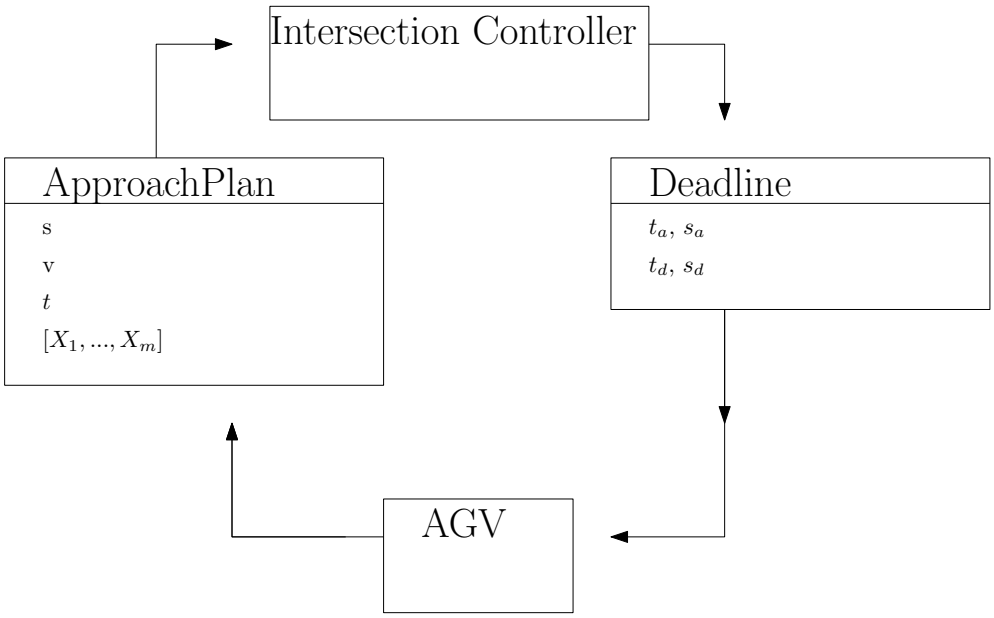


Figure 2.8: Key messages of the "Plan/Deadline" interface governing access to a conflict point intersection.

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 [55] and [56].

2.5.1 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 2.2: 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 2.3: 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.

2.6 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 2.2.

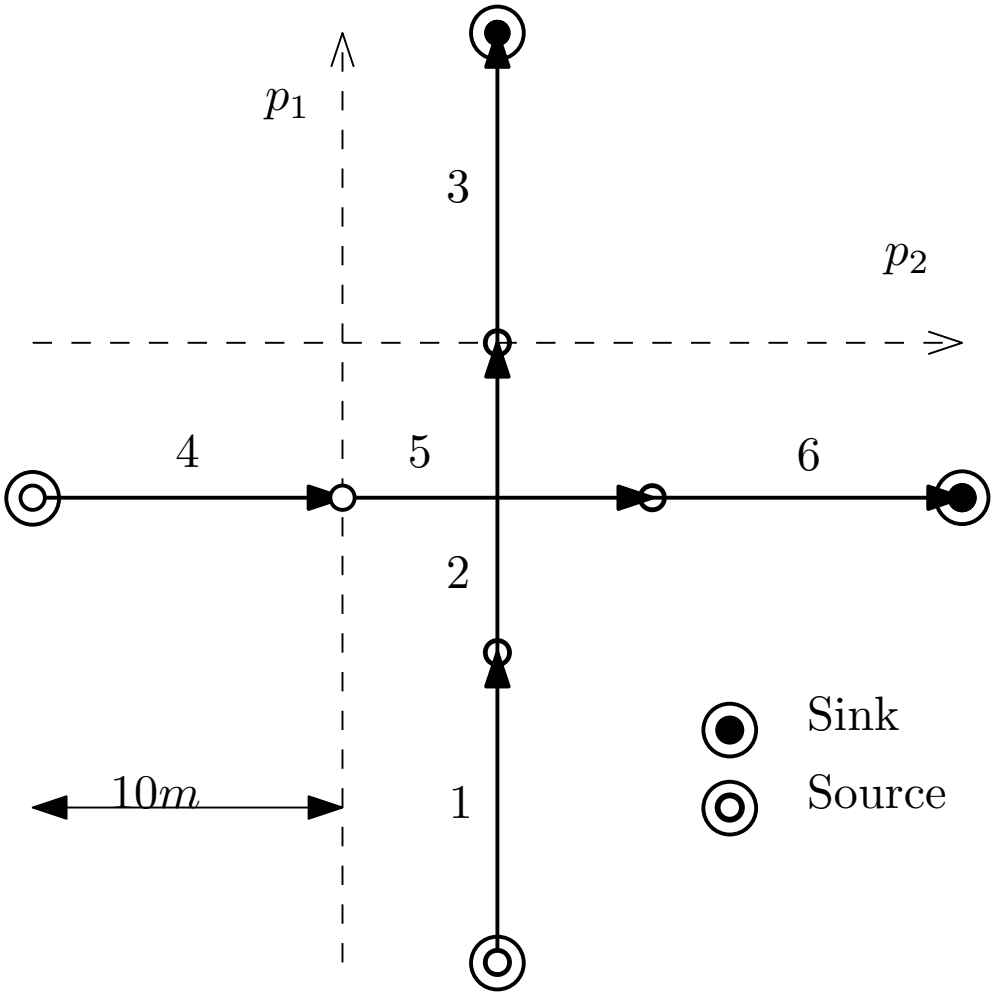


Figure 2.9: Intersection layout with two conflicting routes.

2.7 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 2.9. 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 2.7.

	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 2.4: 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

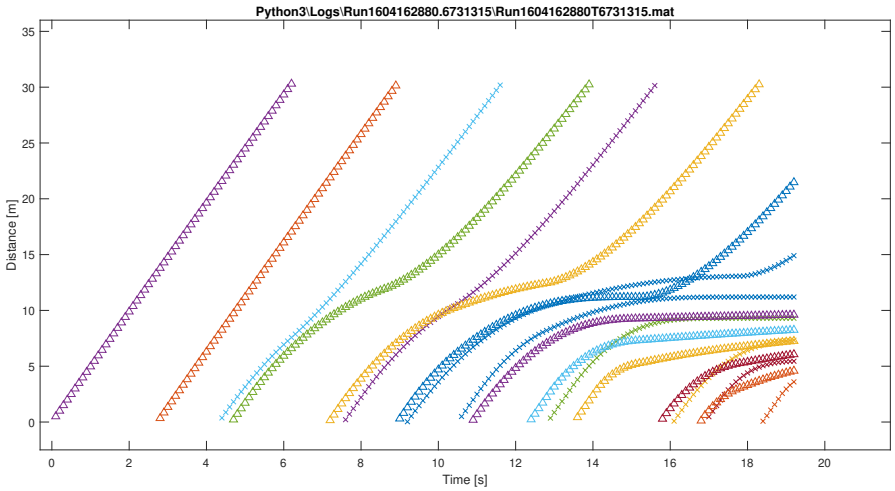


Figure 2.10: Position time series for the semaphore heuristic controller in scenario 1.

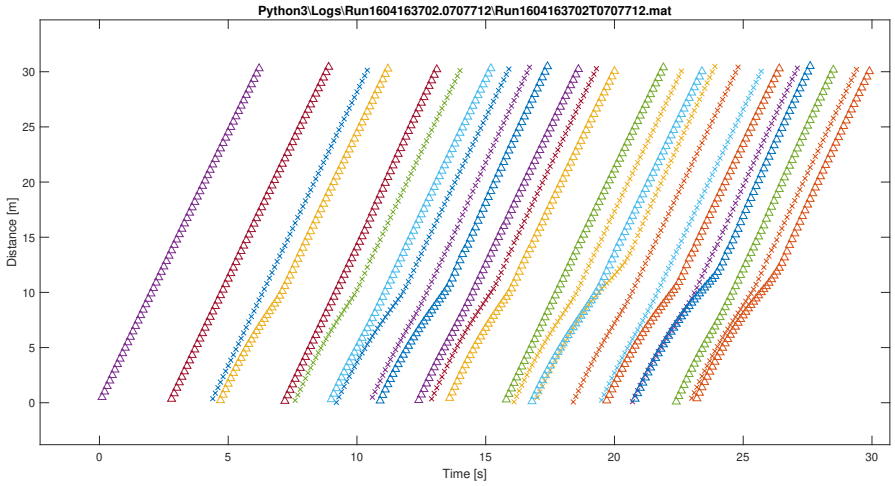


Figure 2.11: The position time series for the FIFO Optimal Controller in scenario 1.

2.7.1 Trajectory Comparison with Fixed Arrival Pattern, 30 Second Run

The distance-time plot for the semaphore controller is shown in Figure 2.10. 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 2.11. 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.

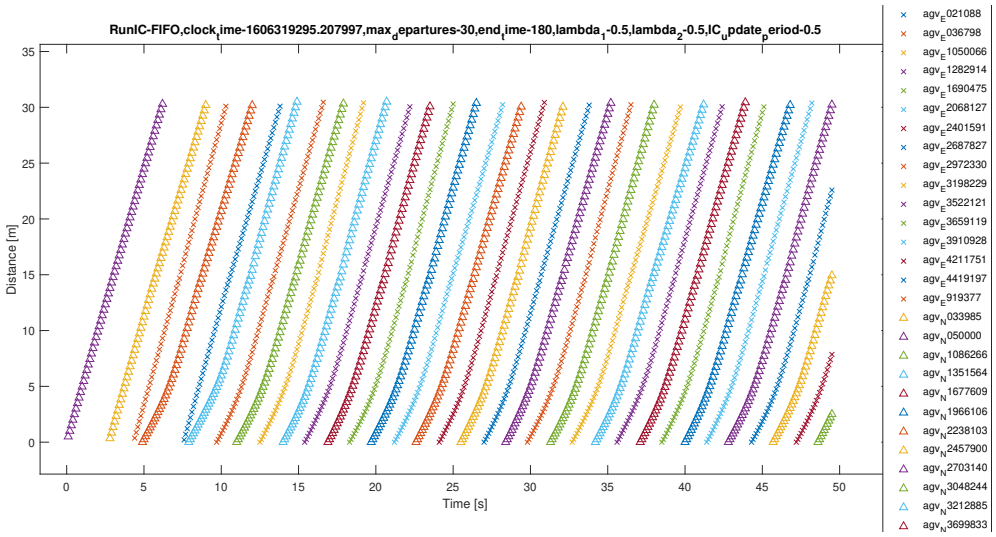


Figure 2.12: The position time series for the FIFO Optimal Controller in the High Traffic High Latency Scenario

In Figure 2.11 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.

2.7.2 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

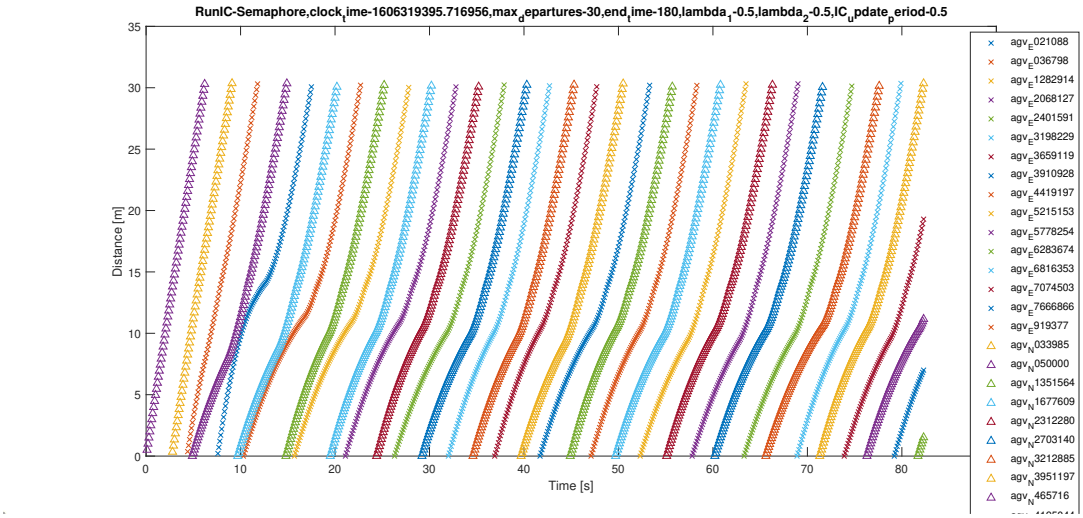


Figure 2.13: The position time series for the Semaphore Controller in the High Traffic High Latency Scenario.

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.

2.7.3 Energy Consumption

The use of an approach speeds minimising total travel time increases the energy consumption per vehicle as shown in Table 2.7.3. 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 2.5: Energy and Completion Time Results

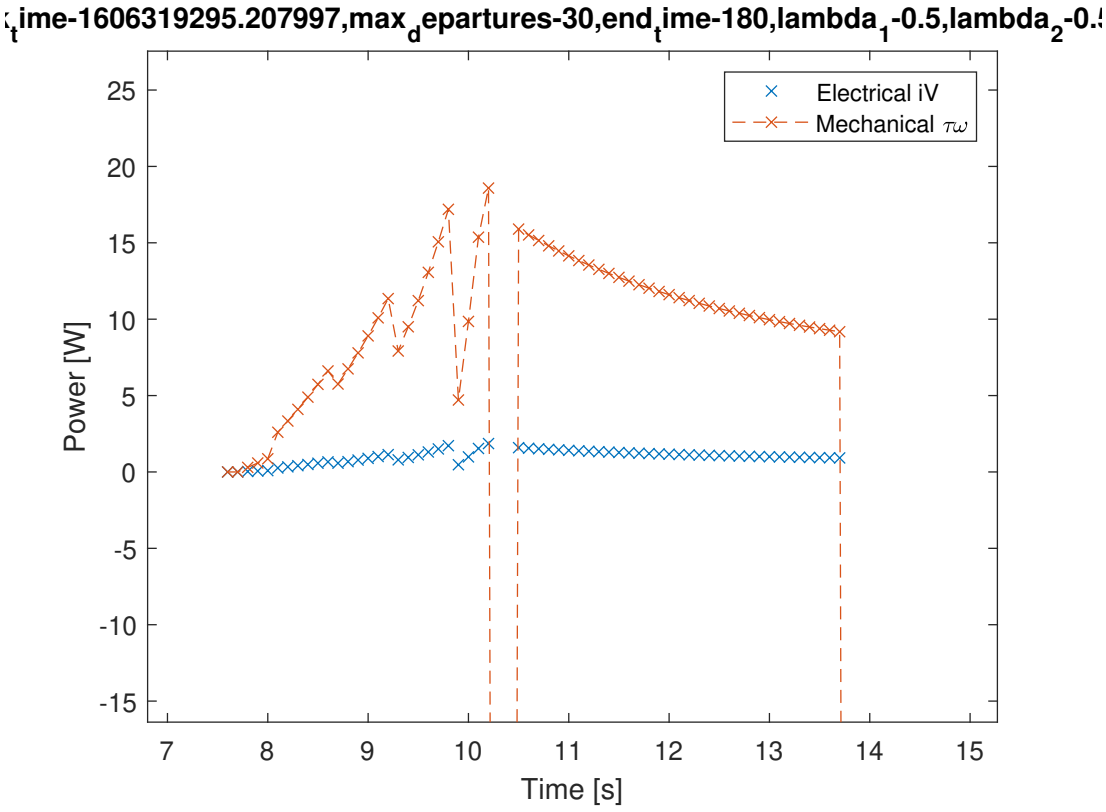


Figure 2.14: The power dissipated over time for one AGV under FIFO control.

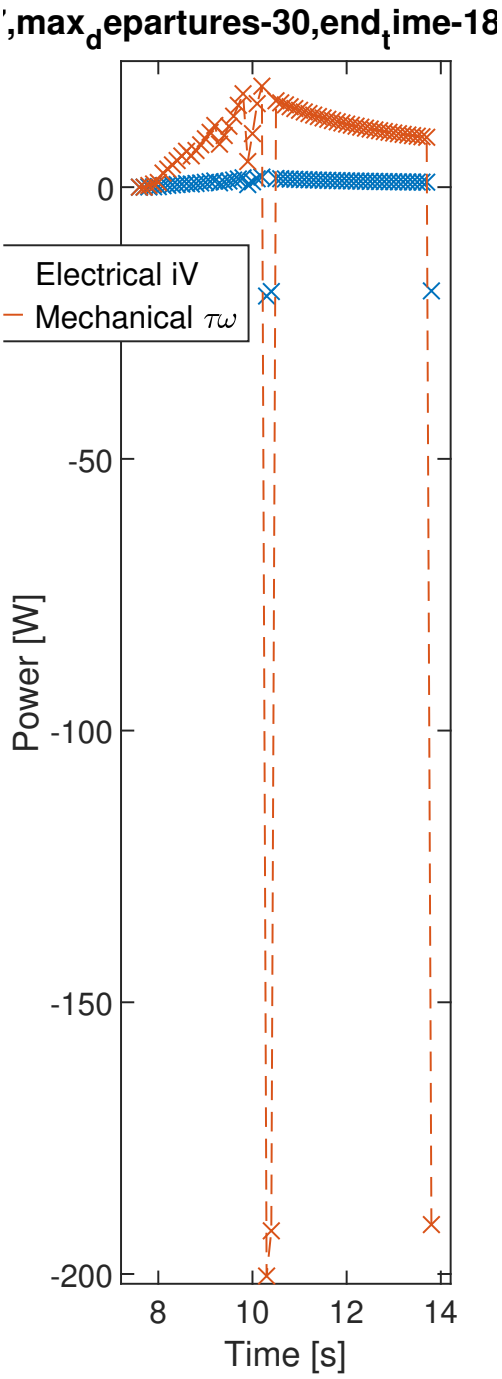


Figure 2.15: Spurious data points of the power dissipated over time for one AGV under FIFO control.

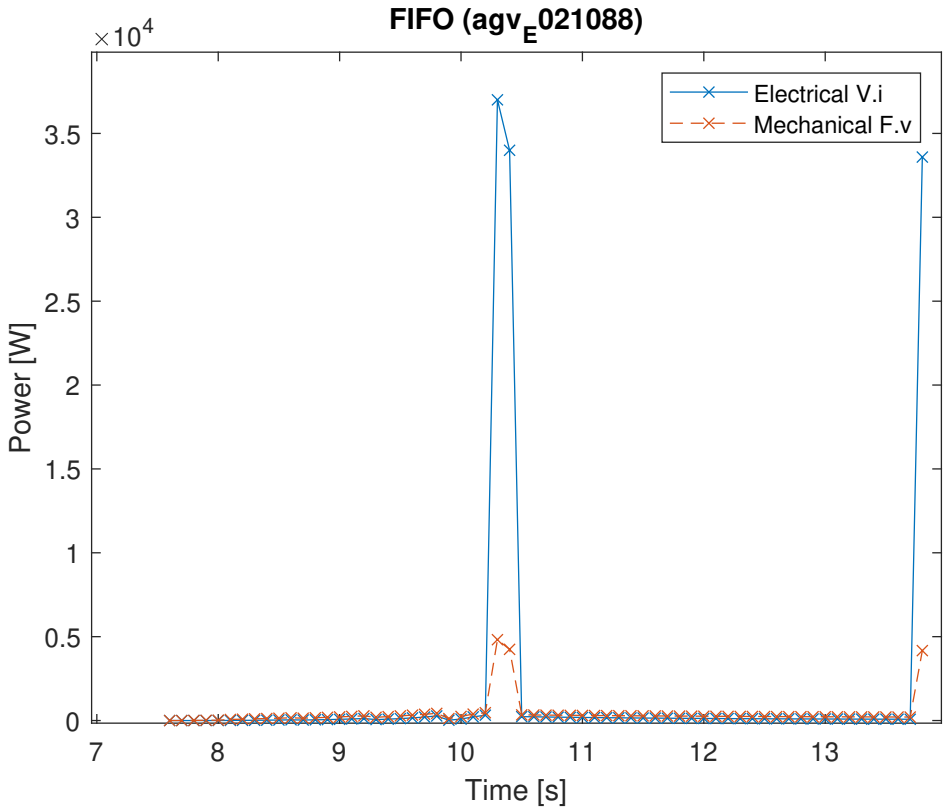


Figure 2.16: The electric power was recalculated from the logged value of armature current i^2R with a resistance of R=0.001ω. The mechanical power weas recaluclated using the abs(motive_force*velocity).

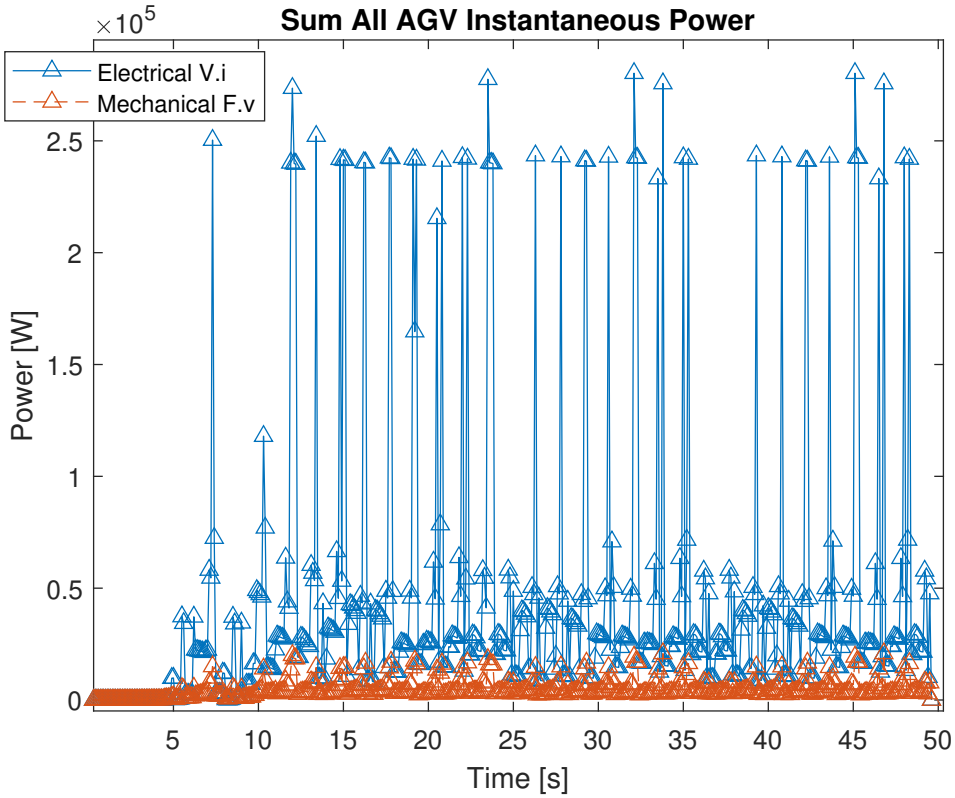


Figure 2.17: The power dissipated over time for all AGVs in total under FIFO control.

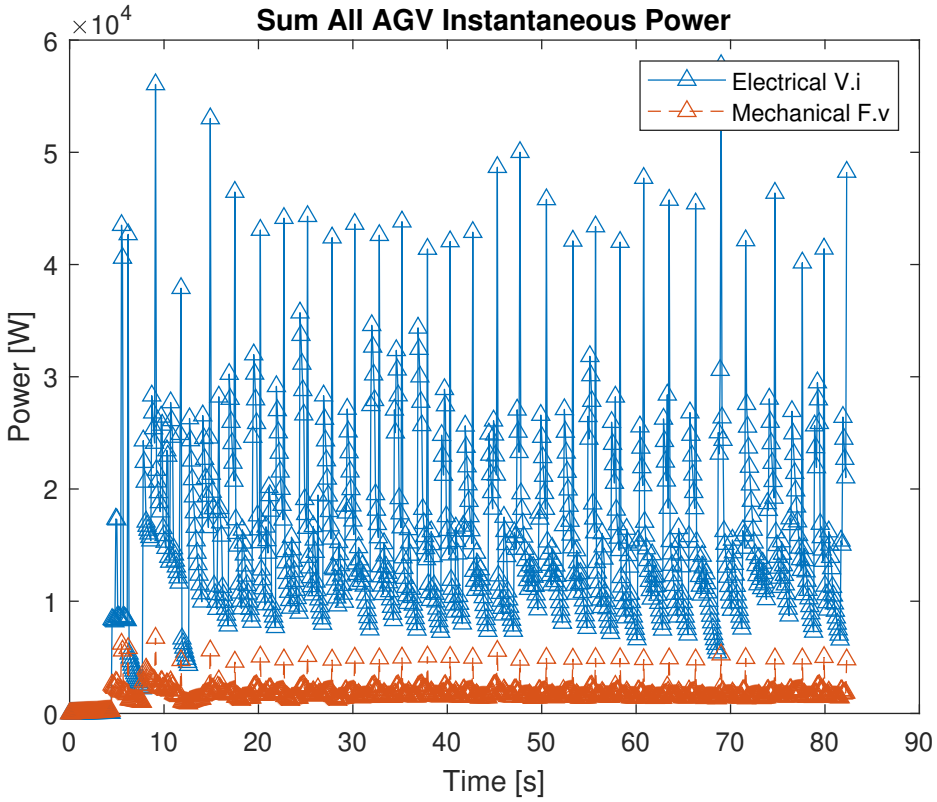


Figure 2.18: The power dissipated over time for all AGVs in total under Semaphore control.

Chapter 3

Fitting Smooth Arcs to Polygon Regions - Comparative Results

3.1 Introduction

To understand the advantages of using clothoid pieces to join two poses in a 2D obstacle field, it is informative to compare the output paths with an alternative curve type. One well known option is to use polynomial curves of degree 3.

3.2 Methodology

The paths are compared on two simulated environments. One is dimensioned for a small AGV with turning radius 0.5m, as might be used to deliver small items in a flexible manufacturing environment. The unexpected obstacle completely blocks the path to the right as in Figure **??**. The second could represent a fork lift type AGV with a larger turning radius of 2m, which is collecting standard size (1m x1.2m) pallets from a storage area. See Figure **??**. The pallets are placed by human drivers, and a reliable system fo detecting their position and orientation is assumed to be in place. The AGV must manoeuvrer to the pose of a target pallet, without colliding with the others. Lower curvature and sharpness allow the AGV to traverse the path faster without compromising load stability.

3.3 Algorithm

Both methods decompose the problem into topology followed by curve fitting. The topology problem is a posed as a directed graph with weighted edges. There is a node for every intersection of the boundary between two regions. Nodes within the same region are fully connected. The weight of each edge corresponds to the euclidean distance between the two nodes. The A* Algorithm is used to search for the set of edges which give the minimum sum of weights between any start and end pose.

The sequence of edges is then used to populate the matrix $\boldsymbol{H}^{(R\times P)} \in [0, 1]$. This is a binary matrix containing R columns, one for each of the R polygonal regions which comprise the accessible space. Each row corresponds to one of the P path pieces and contains a single non-zero element indicating the region to which is assigned. A path piece may be present in more than one region as the regions may be overlapping, but it must always remain completely inside its assigned region.

3.3.1 Polynomial Method

The problem specification calls for a path which changes smoothly in x, y, heading and curvature. This should start at a specified x_s, y_s, ψ_s with zero curvature and end at x_g, y_g, ψ_g with zero curvature.

$$\begin{aligned} x(t) &= a + bt + ct^2 + dt^3 \\ y(t) &= e + ft + gt^2 + ht^3 \end{aligned} \tag{3.1}$$

There is a unique solution for a cubic spline of with fixed (x, y, heading) at the start and goal, passing through fixed x, y positions numbering n. A cubic spline defined by Equation 3.1 has eight free parameters per segment. To give a unique solution, eight constraints are must be found for each segment. Passing through the n waypoints at the end of each segment gives two, one for the x coordinate and one for y. Enforcing continuity of position between the end of each segment and the next leads to two more. Four more can be determined from continuity in the first and second derivative of position for a total of eight.

However, only four constraints are needed at the start and end of the spline. Fixing the final position and heading, the acceleration must be left free. Stacking the parameters into a vector

$$\boldsymbol{p}_i = [a_i, b_i, c_i, d_i, e_i, f_i, g_i]^T \tag{3.2}$$

and

$$\boldsymbol{p} = [\boldsymbol{p}_1, \cdots, \boldsymbol{p}_i, \cdots, \boldsymbol{p}_{n-1}]^T \tag{3.3}$$

leads to the system of linear equations is given in Equation 3.4.

$$[\boldsymbol{A}|\boldsymbol{b}_x] = \left[\begin{array}{ccccccc|c} \boldsymbol{A}_0 & & & & \cdots & 0 & \boldsymbol{b}_{x0} \\ \boldsymbol{0} & \boldsymbol{A}_1 & & & \cdots & 0 & \boldsymbol{b}_{x1} \\ \vdots & & \ddots & & & \vdots & \vdots \\ 0 & & \cdots & \boldsymbol{A}_i & \cdots & 0 & \boldsymbol{b}_{xi} \\ \vdots & & & & \ddots & & \vdots \\ 0 & \cdots & & & & \boldsymbol{A}_n & \boldsymbol{b}_{xn} \end{array}\right] \tag{3.4}$$

Where

$$[\boldsymbol{A}_0|\boldsymbol{b}_{x0}] = \left[\begin{array}{cccc|c} 1 & 0 & 0 & 0 & x_0 \\ 0 & 1 & 0 & 0 & \cos\phi_0 \end{array}\right] \tag{3.5}$$

$$[\boldsymbol{A}_i|\boldsymbol{b}_{xi}] = \left[\begin{array}{cccccccc|c} 1 & 1 & 1 & 1 & 0 & 0 & 0 & 0 & x_i \\ 1 & 1 & 1 & 1 & -1 & 0 & 0 & 0 & 0 \\ 1 & 1 & 2 & 3 & 0 & -1 & 0 & 0 & 0 \\ 0 & 0 & 2 & 6 & 0 & 0 & -2 & 0 & 0 \end{array}\right] \tag{3.6}$$

$$[\boldsymbol{A}_n|\boldsymbol{b}_{xn}] = \left[\begin{array}{cccccccc|c} 0 & 0 & 0 & 0 & 1 & 1 & 1 & 1 & x_n \\ 0 & 1 & 0 & 0 & 0 & 1 & 2 & 3 & \cos\psi_n \end{array}\right] \tag{3.7}$$

The $\boldsymbol{p}_x$ parameters to fit a set of n waypoints can be found by computing $\boldsymbol{p}_x = \boldsymbol{A}^{-1}\boldsymbol{b}_x$. The $\boldsymbol{p}_y$ parameters can be found almost identically, by computing $\boldsymbol{p}_y = \boldsymbol{A}^{-1}\boldsymbol{b}_y$. Constraint vector $\boldsymbol{b}_y$ is instead constructed using the y coordinates of the waypoints in b_i and the sin of the start and end heading in $\boldsymbol{b}_{y0}$ and $\boldsymbol{b}_{yn}$.

3.4 Optimisation

The point to point solution through the region intersection points is quite a good solution. Continuous in $\dot{X}$ and $\ddot{X}$, it reaches the goal position and heading. It is excessively constrained by the requirement to meet the intermediate waypoints. Using the optimisation in Equation 3.8

$$\begin{gathered} \min \textstyle\sum_{j=1}^{N} ||\frac{dP_j^3(t)}{dt^3}||^2 \\ \text{subject to} \\ P_j(1) = P_{j+1}(0) \\ \dot{P}_j(1) = \dot{P}_j(0) \\ \ddot{P}_j(1) = \ddot{P}_j(0) \\ P_1(0) = X_S \\ P_N(1) = X_G \\ \dot{P}_1(0) = \dot{X}_S \\ \dot{P}_N(1) = \dot{X}_G \\ \text{also subject to} \\ \boldsymbol{H}_{j,r} \Rightarrow X_r^{min} \leq P_j(t) \leq X_r^{max} \end{gathered} \tag{3.8}$$

The implementation of the region constraint in Equation 3.8 must ensure the entire arc remains inside the assigned region according to $\boldsymbol{H}$. For piecewise linear arcs this can be done by checking the containment of the start $P_0(0)$ and end $P_N(1)$. Got cubic splines it is more challenging. Deits and Tedrake **??** describe an approach based on

Sum of Squares, where a small Semidefinite Program is solved in the parameters of P_j, to avoid sampling at different t values, which can lead to the path cutting corners and even passing through thin obstacles.

First we notice that the constraints on one segment from its axis aligned containing region described by $x_min, x_max, y_min, y_max$ can be written as vector inequality

$$q(t) = \begin{bmatrix} x_{max} - a - bt - ct^2 - dt^3 \\ a + bt + ct^2 + dt^3 - x_{min} \\ y_{max} - e - ft - gt^2 - ht^3 \\ e + ft + gt^2 + ht^3 - y_{min} \end{bmatrix} \geq \mathbf{0} \forall t \in [0, 1] \tag{3.9}$$

. The condition in Equation 3.9 can only hold if and only if it can be rewritten in the form

$$t\sigma_1(t) + (1 - t)\sigma_2(t) \tag{3.10}$$

. This leads to

$$\sigma_1(t) = \begin{bmatrix} x_{max} - a - b - ct - dt^2 \\ a + b + ct + dt^2 - x_{min} \\ y_{max} - e - f - gt - ht^2 \\ e + f + gt + ht^2 + -y_{min} \end{bmatrix} \tag{3.11}$$

and

$$\sigma_2 = \begin{bmatrix} x_{max} - a \\ a - x_{min} \\ y_{max} - e \\ e - y_{min} \end{bmatrix} \tag{3.12}$$

. The standard Sum of Squares approach calls for collecting the parameters of $\sigma(t)$ by the order of t. Matching coefficients against

$$\sigma_i = \beta_1 + \beta_2 t + \beta_3 t^2 \tag{3.13}$$

, gives

$$\beta_1 = \begin{bmatrix} x_{max} - a - b \\ a + b - x_{min} \\ y_{max} - e - f \\ e + f + y_{min} \\ x_{max} - a \\ a - x_{min} \\ y_{max} - e \\ e + y_{min} \end{bmatrix} \tag{3.14}$$

and

$$\beta_2 = \begin{bmatrix} -c \\ +c \\ -c \\ +c \\ 0 \\ 0 \\ 0 \\ 0 \end{bmatrix} \tag{3.15}$$

and

$$\beta_3 = \begin{bmatrix} -d \\ +d \\ -d \\ +d \\ 0 \\ 0 \\ 0 \\ 0 \end{bmatrix} \tag{3.16}$$

. Where the parameters for σ_2 have been stacked after those from σ_1.

In order for σ_i to be a sum of squares, there are the following conditions on β_1, β_2, β_3:

$$4\beta_1\beta_3 - \beta_2^2 \geq 0 \beta_1, \beta_3 \geq 0 \tag{3.17}$$

In the simple case where the regions are axis aligned, this immediately creates a problem as the first two equations have $\beta_3 = d$ and $\beta_3 = -d$. The only value of d which

will satisfy the sum of squares condition is zero. However the initial guess which joins the corner points, satisfies the region occupancy constraints with a non-zero d.

For this reason the constraints are enforced using 50 samples for each path piece. This leads to a small degree of corner cutting which must be included in the safety factor used on the vehicle body width used to inflate the obstacle. It also results in a large number of constraints as there are four for each sample. For this to be possible we must use a fixed spatial sampling length rather than a fixed number of samples per piece as path pieces can vary in length substantially which makes the degree of corner cutting variable.

3.4.1 Clothoid Method

Initialization with parameters which meet the continuity and obstacle constraints with clothoid pairs proved to be a challenging task. This is expected to improve the convergence time. To generate a set of parameters to meet continuity constraints for initialization of the region method is very similar to the interpolation problem addressed by Bertolazzi et al (2018) [**?**]. The main difference is that the intermediate waypoint headings are free variables in the initialization, unlike in interpolation.

If the waypoint headings are fixed, and curvature at them to zero, the clothoid interpolation problem can be solved for each segment independently, as described by [57]. Different point to point methods could be used such as the bisection method [58], geometry [59] and minimax sharpness [11]. Lacking a natural heading, a heuristic based on the maximum squared sum of $a^2 + b^2$ and requiring $a > 0$ and $b < 0$ where a is the distance to the intersection point from P_1 and b is the distance from P_2. The different sign is important for the direction of travel, to put the intersection in front of the first pose and behind the second.

The multiple shooting clothoid method uses additional parameters for the start pose of every segment. If this was initialized to zero, interior-point method reached convergence in about 70 seconds in environment 2. None of the more involved methods improved on this so we used the simple approach for further testing.

Figure 3.1: Convergence time variation with weighting on item fetch environment

b	1×10^{-3}	1	100	1000*
fevals	7929	10873	11986	200,000*
execution time (s)	55	72	80	1345*
path length (m)	7.071	11.694	36.038	5.695*

*Convergence Failure

Table 3.1: Convergence for different b values for the multiple shooting approach, in pallet environment

3.5 Convergence dependence on b

In environment 1 both large and small values of w led to poor convergence. One possible explanation is numerical conditioning being damaged by the weighting. A large w results in a solution with small α, while the length remains on the same order. Some matrix operations may return inaccurate results with different magnitude parameters. To test the hypothesis that only method using dense matrices such as sqp would be affected, the effect on the interior point method (which does not depend on dense matrix operations) is reported in Table 3.5. This shows that changing the weighting damages convergence significantly to the extent that no valid result is found even after 200000 function evaluations. There were no warnings printed by fmincon during this run. The value of feasibility was positive of the order 1e-6 and decreasing very slowly, while the objective function was large of the order 1e6 and also decreasing very slowly (relative to its magnitude).

Contrary to expectation, reducing b to emphasize the path length reduced the number of iterations to convergence. Table 3.5 shows an inverse relationship between b and

the number of iterations. Sufficiently large b makes the problem more difficult to solve as the path segments deflect further, making a linear approximation worse. This effect is indirect because the constraints use the precise nonlinear dynamics and the interior-point method does not solve a series of linear approximations to the constraints like sqp. Interior point should always converge if given an initial guess in the feasible reason, unless the problem is non-convex. The initial graph step intended to remove non-convex elements of the problem may lead to suboptimal paths as the segments deviate from straight lines. But this does not explain the increasing number of iterations.

Curvature is more weakly linked to the parameters than the total length. The curvature objective is the squared sum of $2r$ parameters while the length objective is the squared sum of $4r$ parameters. Both of them leave numerous parameters to be coupled through the constraints as in total there are $9r$ parameters for the multiple shooting formulation.

3.6 Generating a Convex Region Representation from LIDAR Data

A widely used representation for a 2D obstacle field is an occupancy grid [**?**]. Also see Probabilistic Robotics textbook [**?**] for details on creating a map from sensor data. Each cell represents an area of the floor, with a number $p \in [0, 1]$ indicating the probability it contains an obstacle. Thresholding can be used to create a binary map of occupied and unoccupied cells. In order to connect adjacent unoccupied cells to form regions numerous approaches from image processing are available. A method suitable for simple environments is vertical cell decomposition [**?**]. This begins with piecewise linear polygons, which can be created by connecting the cell corners of a binary occupancy grid.

3.7 Test Environment

The test environment was created for an automated fork lift AGV research platform which is taken to be representative [**?**]. The obstacles are based on (1.0×1.2)m pallets which are commonly used in the UK and Netherlands [**?**]. The datasheet for the manually operated vehicle on which the AGV is based, a Hyster E30-40HSD gives the

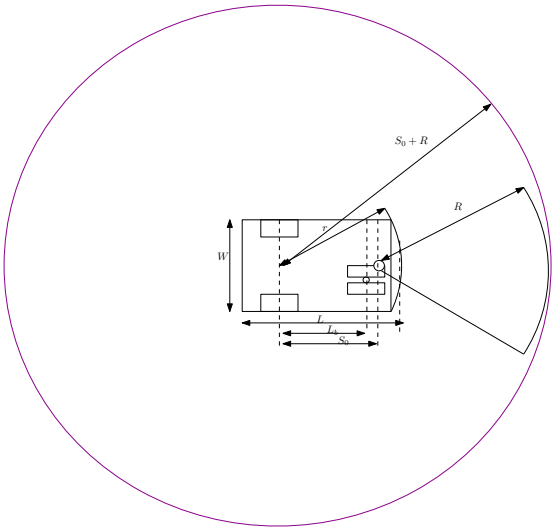


Figure 3.2: Dimensions used to expand the obstacles

dimensions in Table 3.7. The dimensions are drawn on a plan of the vehicle in Figure 3.7.

The datasheet gives the maximum speed as 7.2 miles per hour (3.22m/s). Traction is provided by dual 4.8kw motors. The unloaded weight of the vehicle is 3059kg and the battery 1043kg for a total of 4201kg.

For correct operation it is important to consider the exclusion zone of the safety rated range sensor fitted to the front of the vehicle. If an obstacle breaches the exclusion zone the AGV must perform an emergence stop, or in some cases slow down significantly.

Parameter	Dimension (mm)
W	1067
L	1583
r	1289
L_s	1001
S_0*	1200
R*	1300

Table 3.2: Dimensions from datasheet. *Stopping distance R based on top speed 3.22m/s and hypothetical braking deceleration of 4m/s^2

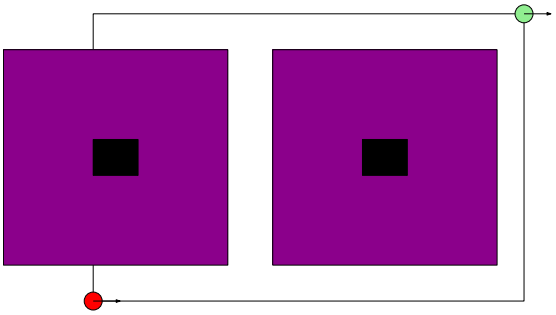


Figure 3.3: Pallet environment. Obstacles are black with expansion by the vehicle disk shown in purple.

To avoid slowing down the path planning must account for not only the shape of the vehicle but also the shape of this zone. Often this is a cuboid slightly wider than the vehicle, sufficiently long that the AGV can come to a complete stop from full speed before the front makes contact with a static obstacle. More details are available in the NIST Safety Standards [**?**]. To avoid

In the two simulated environments the obstacle field is represented in 2D. The bounding circle dimension is strongly influenced by the stopping distance R. Starting from the constant acceleration equation $v^2 = u^2 + 2as$ and setting $v = 0$ gives the stopping distance $s = \frac{u^2}{-2a}$.

Expanding the environment with a disc the size of the largest dimension of the vehicle is a gross approximation, which will lead to over cautious paths. It would be better to represent the obstacles in $(2 + 1)$ dimensions, and plan in (x, y, ψ). Obstacles can then be expanded based on the current heading of the path as described in [**?**] who use a sum of squares approach to find (2+1)D convex polygons.

Chapter 4

Conflict Points at Intersections and Adaptive Paths

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 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 [60].

A systematic procedure computing the conflict points in an intersection is given in [61]. 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 [39]. 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 [62]. 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 con-

flicts which occur and roundabouts. These are reviewed in [39]. A centralized solution with an intersection manager minimizing delay and energy consumption is described in [63]. 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 [64], 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 [65]. Another work investigating virtual lanes is [66]. 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 [67] 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 [50]. 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 [50], 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 [68]. 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 [68] is shared with many earlier works regarding coordination of Unmanned Arial Vehicles, many of which relax the assumption of static paths. In this way [69] 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 [70]. An earlier theoretical treatment based on iterative bargaining with soft collision constraints is given by [71]. 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 [72], per-vehicle optimizations runs simultaneously to reduce execution time. This ensures recursive feasibility and closed loop stability. Another DMPC approach is given by [73]. 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.

The distributed consensus on the arrival order underpins the distributed solution for trajectory planning. The body of work considering flexible paths for aircraft relies on similar techniques to the latest works targeted at Connected Automated Road Vehicles where the paths are fixed. Specifically [67] uses sequential per-vehicle optimisation to find the highest safe speeds subject to static obstacle constraints, and the trajectories of vehicles earlier in the sequence. The crossing order is determined by the arrival time at the 'point of no return' according to the motion model of each vehicle. The high frequency closed loop controller on one vehicle continues to operate while the others are formulating their own trajectory plan. Similarly the work on adaptive paths in [73] uses simultaneous per-vehicle optimisation to find the trajectory as a sequence of control actions and associated positions at regular time intervals which the vehicle is predicted to occupy up to a receding time horizon. Both offer a suboptimal global (all-vehicles) solution, which is guaranteed to be safe. Could the simultaneous approach improve performance of DMPC? Is it just a different name for the same algorithm? An existing constrained optimisation based path planner could be used to generate trajectories by assuming a simple speed profile. Mutual constraints between two trajectories can then be applied, with one constraint for each time sample, leading to a solution to the central(all-vehicle) multi-vehicle trajectory planning problem. The local(per-vehicle) optimization is identical to the global(all-vehicle) one, only the passing order choice need to reach consensus. Either of the two aforementioned decentralized ordering algorithms should be applicable. Which is most promising, resulting in a consensus with the least deviation from central(all-vehicle) optimality? Does the answer change depending on the nature of the path optimisation, should it be based on a clothoid spline or radial polynomials?

Appendix A

Optimal Smooth Paths Based on Clothoids for Car-like Vehicles in the Presence of Obstacles

Paper attached.

International Journal of Control, Automation and Systems 19(X) (2021) 1-20
http://dx.doi.org/10.1007/s12555-020-0179-1
ISSN:1598-6446 eISSN:2005-4092
http://www.springer.com/12555

Optimal Smooth Paths Based on Clothoids for Car-like Vehicles in the Presence of Obstacles

Edward Derek Lambert*, Richard Romano, and David Watling

Abstract: Automated Guided Vehicles are increasingly used for material transfer in factory and warehouse environments amongst humans and human operated vehicles. Safe and efficient operation is challenging when there is a mix of human and automated traffic as fixed guide paths can become blocked more frequently. In this work we aim to show smooth and efficient paths based on clothoid curves can be used to automatically plan diversions which can be traversed at high speed by automated fork-lift vehicles, which are car-like in the sense they have a limited turning radius and angular acceleration. The approach, based on numerical optimisation within convex region constraints is described in detail, and numerical results for curvature and sharpness are compared to a cubic spline on a small number of simulated environments. The clothoid spline is less affected, in terms of its objective function, by a shift in the obstacle boundaries than a cubic spline, for obstacle shifts below 0.5m. The clothoid spline takes longer to converge for but the output path has attractive qualities like lower peak sharpness, enabling high speed operation. The method is therefore most useful for applications where path quality is important and updates are required less frequently. Changing the objective function by increasing weighting parameter b allowed the path shape to be tuned to reduce the peak sharpness, at the cost of increasing the total length. With $b > 100$, convergence was poor because parts of the spline were pushed outside the assigned region, an artefact arising from the constraints only being enforced at the start and end of each segment. The analytical Jacobian of the constraints was effective at reducing the number of function evaluations to reach convergence.

Keywords: Continuous curvature path planning, convex regions, nonholonomic car-like vehicle, non-linear, obstacle avoidance, optimal path.

1. INTRODUCTION

Consider a fleet of Autonomous Guided Vehicles (AGV) moving material in an automated manufacturing plant. A lattice roadmap made up of virtual guide paths is a widely used solution for planning the motion of each AGV [1]. This is designed by engineers installing the AGV system, and may remain in use for many years. In environments which are shared with humans and human operated vehicles there is a greater probability of unexpected obstructions blocking the guide paths. Automatic replanning to avoid these obstructions has the potential to increase performance and robustness of shared environment AGV systems. Although numerous techniques have been developed for planning paths of car-like vehicles around obstacles [2], none has achieved wide acceptance in industry [3, 4]. The problem addressed in this paper is finding a smooth path around obstacles which can be followed exactly by a vehicle with car-like dynamics.

It is important to consider the variety of solutions which have been developed already in this area. A range of techniques for motion planning are well described with examples in [5] including graph search methods such as Probabilistic Roadmaps with Dijkstra in addition to incremental search methods like Rapidly Exploring Random Trees. Katrakazas *et al.* [6] goes into more detail for only those techniques suitable for on-road autonomous vehicles and Paden *et al.* [2] gives optimality and completeness results for many in a handy comparison table. Recent developments have used gradient descent to modify Bezier curves based on obstacle keypoints [7], evaluated alternative clothoid tentacles [8] and found the parameters of interpolating clothoids as an optimisation [9]. The benefits of clothoids for controlling lateral acceleration identified in [10] can be exploited by unoccupied vehicles, which can travel faster on smooth paths without lateral instability [11, 12].

Path planning techniques can be divided into spatial sampling based and continuous methods. Sampling based planning algorithms operate on a discretization of the

Manuscript received March 13, 2020; revised August 18 2020; accepted September 8, 2020. Recommended by Associate Editor Son-Cheol Yu under the direction of Editor Euntai Kim.

Edward Derek Lambert, Richard Romano, and David Watling are with the Institute for Transport Studies, University of Leeds, 34-40 University Road, Leeds LS2 9JT U.K. (e-mails: {tsedl, R.Romano, D.P.Watling}@leeds.ac.uk).
* Corresponding author.

state space. Into this category fall roadmap planners such as [13] where the graph resulting from discretization is reused multiple times. The graph for adaptive paths is likely to be used only once as the obstacle field is likely to change based on recent sensor data. Roadmap planners are still extremely useful at the global scale where local sensor updates are less relevant. One result is the split architecture described by [2] where a roadmap planner is used for strategic planning (planning over a longer time scale like a few minutes, extending from one part of the site to the other) and a different approach is used for tactical planning (creation of a detailed trajectory for the next few seconds). In [14], this approach, using different techniques for the two time-scales is described as integration planning.

There are other sampling based methods which can also be useful for path adaptation where local sensor updates are important such as Dense Random Trees as described in [15,16] where the discretization is performed as the search proceeds. Sampling is frequently used to make problems of high dimensionality feasible, but can only offer resolution completeness. This is the guarantee that if a solution exists at the sampling resolution, then it will be found. Sampling from configuration space can result in paths which must be smoothed before they are traversable, so the most relevant techniques are based on sampling from the control space, or using parametrized curves so that every sample is feasible as in [17]. A frequent issue with dense random tree sampling methods is the introduction of artefacts in the solution which are difficult to remove by post-processing.

By contrast, the family of solutions based on numerical optimisation which operate directly on the continuous state space offer improved path quality and guarantees. These methods can be divided into parametric formulations which describe the path as some type of curve such as a polynomial [18] and those where the path is represented by a series of time samples which satisfy the differential constraints such as Timed Elastic Bands [19]. Compared to parametric methods, those optimising over a series of samples must search a much greater number of variables and also account for more constraints. This leads to additional computational burden, so they are often limited to a short time horizon and make use of a reference path to linearise obstacle constraints as the tactical planner in [20] which also uses output constraints to turn overtaking into a convex problem. For longer paths which can be stored in limited memory and reused parametric methods may be preferable, provided they are able to represent the dynamic limitations of the AGV.

Path representations which are suitable for the dynamic constraints of car-like vehicles can be based on different types of spline. Splines which are Cartesian can be calculated conveniently but they are only traversable if polynomial terms up to 5th order are included [21]. Polar splines have a smoothly varying curvature at first order and above but they are unable to represent a straight line they must be mixed with other curve types to form a complete path. Other curve types such as Bezier curves exist but one representation which is particularly suited to industrial AGV roadmaps is the clothoid curve or Euler spiral [22]. Using this parametrization, and constraining peak curvature and sharpness, the resulting path will be feasible for a car-like vehicle at non-zero speed. The importance of sharpness limitation is sometimes overlooked but this is a real physical limit on the motion of a vehicle. This is because the sharpness is proportional to the angular acceleration at a constant traversal speed. Previous work on finding clothoid based paths around obstacles has mostly used spatial discretization to generate a series of points between the origin and destination, followed by curve fitting to find the clothoid segments which best fit to the points. This is a practical solution and variants of it are used by [23] who generates the key-points from a sequence of position samples from a manual drive, and [24] who fits to a series of predefined manoeuvres: u-turn, lane change and so on. As with other sampling based methods, the choice of sample points affects the final solution, leading to suboptimal solutions. This was made clear in [25] where sampling and curve fitting was compared to a direct optimisation method.

An early method for creating continuous curvature paths based on clothoids, arcs and straight lines was the CC-Steer algorithm [26]. This local planner could be used to connect samples from configuration space to create a probabilistic road map. A similar algorithm from around the same time from [27] was also able to create CC-paths without using clothoids by considering curvature continuity while approximating a holonomic path with a heuristic exploiting the differential flatness of car pulling trailers. Differential flatness of a dynamical system indicates the prior states can be determined from the current state with no exogenous variables [28]. Fraichard and Scheuer [26] showed CC-Steer approximated Reeds-Shepp [29] paths which are provably the shortest for connecting points with heading continuity. Using the maximum sharpness and maximum curvature to produce the shortest path is fundamental to the operation of CC-Steer, but this is not the only important objective. Often it is preferable to minimize the sharpness of turns in order to reduce lateral forces and maintain high speed.

The contribution of Henrie and Wilde [24] was to describe an algorithm to join two configurations with the least maximum curvature and least sharpness to create comfortable paths similar to those a human driver would follow. This used symmetric clothoid pairs to assemble paths with the same structure as [26]. One limitation of this work is that the clothoid pairs are always arranged symmetrically which limits the range of manoeuvres. This was addressed by the bisection method proposed by [23]

which performs a numerical search to find either two or four clothoid segments, which are not required to be symmetrical, only matched so that the curvature at the end is zero. Another contribution of [23] is to test one approach for creating smooth global plans by fitting arcs and lines and clothoids to a series of samples from a GPS trace of a test vehicle driven by remote control. Gim [30] goes into further detail in an Appendix B regarding the reachability of clothoid pairs, but does not examine the geometric limits discussed in Section 2.2. The search procedure is fast at finding the parameters which meet the constraints but no algorithm is given for the correct choice of points to interpolate given a set of obstacles.

Recent works searching for paths with limited curvature rate such as [31] who used a superset of clothoids to find the best approximation to a holonomic path with limited curvature and provide a tuning parameter that could be tweaked to avoid obstacles. Solanes *et al*. [32] also approximated holonomic paths but trained a neural network to speed up generation of the initial parameter guess. A double continuous curvature (DCC) path planner is a component in the path tracker described in [12]. The shortest DCC path from the current pose to the global reference path is found every control cycle by Nelder-Mead without considering obstacles. Silvan and Grassi [33] found a compact representation of a smooth road centre line consisting of arcs, lines and clothoids. Existing methods e.g., [34], could then be used to join certain key poses depending whether they would track a roundabout or a straight road. Others such as [35] used a smooth road centre line as a reference to linearise the obstacle constraints and then sequentially minimized maximum curvature and sharpness in addition to deviation from the reference path.

In this paper we present a numerical optimisation formulation which can be used to find a clothoid spline which reaches an arbitrary goal through a series of convex obstacle-free regions. This is distinct from earlier works such as [9] as in this paper only the start and the goal poses are fixed, providing freedom to improve the objective within the natural constraints of obstacle polygons rather than being tied to heuristically selected waypoints. It can also be trivially adapted for point-to-point curve fitting similar to that of [23] but with the resulting path minimizing an arbitrary cost function over its length as described in Section 4.4. The new method is able to take into account constraints arising from obstacles directly, and finds a result using highly optimised off-the-shelf nonlinear optimisation algorithms: Interior-Point Method and Sequential-Quadratic-Programming were tested. Rather than simulate a path following controller with a specific dynamic model, we reproduce results from an existing root finding method for the smoothest (least sharp) path [23] where the clothoid curve outperformed other primitives. The solution presented improves on current methods with a way to find an optimal path directly from the obstacle representation without introducing sampling bias by selecting waypoints. Constraints and soft objectives are separated and a weighting parameter introduced which allows control over the trade off between minimum sharpness and path length. This makes adaptive clothoid paths useful for a multi-AGV site which must have predictable behaviour and high efficiency.

2. MATHEMATICAL REPRESENTATION AND PROBLEM DEFINITION

Clothoid curves are widely used and appreciated for creating smooth drivable paths with limited angular acceleration. They are less frequently used within a numerical optimisation framework, instead many heuristic methods for calculating their parameters have been developed. In what follows we look at the properties of clothoids which are relevant to numerical optimisation.

2.1. One clothoid segment

The clothoid is defined as a curve whose curvature κ increases linearly along its length. The rate of increase is called the sharpness $\alpha = d\kappa/ds$. Points on such a curve are defined by two parameters; the arc length s and the sharpness α. These two parameters describe a curve spiralling out of the origin along the x axis towards one asymptote with infinite positive curvature (turning anticlockwise) in the positive $x-y$ quadrant for $s > 0$, $\alpha > 0$, and towards a negative asymptote with infinite negative curvature in the negative $x-y$ quadrant where heading is decreasing (turning clockwise) for $s < 0$ or $\alpha < 0$. The negative distance curve is just a reflection of the positive one and all arcs can be joined by appropriate rotation and translation so we may fix $s \geq 0$. The positive part of the curve with sharpness $\alpha = 5$ is shown in Fig. 1. It can be evaluated in Cartesian coordinates with the Fresnel integrals which are reproduced in (3)-(4). The change in angle over one segment is the deflection δ.

$$\kappa = \kappa_0 + \alpha s, \tag{1}$$

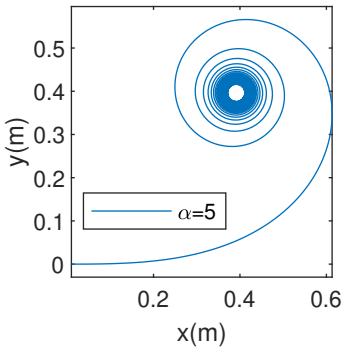


Fig. 1. Clothoid spiral with $\alpha = 5$ for $s > 0$.

$$\delta = \int_0^s (\kappa_0 + \alpha u) du = \kappa_0 s + \frac{\alpha s^2}{2}, \tag{2}$$

$$x = C(\alpha, s, \kappa_0) = \int_0^s \cos(\kappa_0 + \alpha u) du, \tag{3}$$

$$y = S(\alpha, s, \kappa_0) = \int_0^s \sin(\kappa_0 + \alpha u) du. \tag{4}$$

The symbol ψ will be used for resultant heading angle after a number of segments. Whereas each δ increases unbounded, ψ is an angle measured clockwise from positive x direction and may be wrapped in the range $[0, 2\pi]$ without consequence. The configuration of a rigid body in a 2D plane with components $[x, y, \psi]$ will be referred to as a pose. See Appendix B for more details. The range of poses reachable by varying the parameters of a single segment are limited. Any x and y position can be reached by choice of parameters α and s - this makes intuitive sense as the parameter space is two dimensional as are the constraints. In order to meet heading ψ and curvature κ constraints as well, a spline composed of multiple clothoid segments is needed. Note that a clothoid segment with $\alpha = 0$ will form either a straight line of length s if the initial curvature is zero or an arc of length s otherwise.

2.2. Required number of Segments for Interpolation

First we consider G^2 interpolation with clothoids. This involves fitting clothoid segments to a series of points with fixed $[x, y, \psi, \kappa]$ as addressed in [9, 36]. It is helpful to understand the way the required number of clothoid segments varies depending on the constraints applied. As we intend to optimise some objective function of the curve it is required that the solution is under-determined, that numerous feasible solutions to the interpolation problem exist allowing us to search over them to find the best.

G^2 continuity is needed for a smooth path which is traversable for a car-like vehicle [23, 24]. Its importance for fork lift operation is detailed in [12]. As explained in Section 2.1 there are two additional degrees of freedom available for each clothoid segment included in a G^2-continuous spline. The requirement to end on a specified point with $[x, y, \psi, \kappa]$ provides four constraints. This implies that there is a unique solution for two segments per point as this gives rise to four parameters and four constraints.

In order to connect a clothoid segment with a straight line keeping G^2 continuity, the clothoid must have zero curvature at the point of connection. A clothoid pair with zero curvature at the end can be described as 'matched': The curvature change $\kappa = \alpha \cdot L$ over each segment is equal and opposite. Such a pair is shown in Fig. 2 alongside a curvature-arc-length plot for illustration. Gim et al [23] show that the minimum reachable heading with two matched clothoids is given by (5).

$$\psi > \phi = \arctan\left(\frac{y}{x}\right). \tag{5}$$

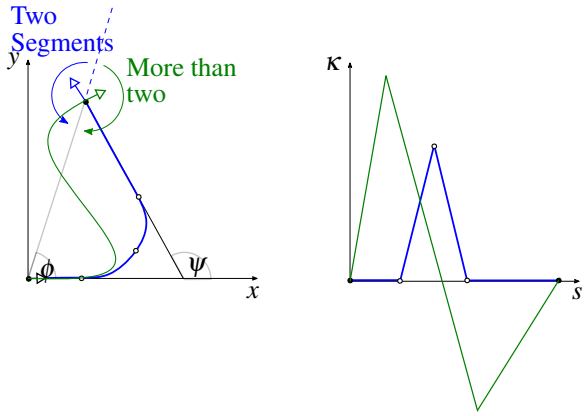


Fig. 2. The angle ϕ of the line joining the origin with the target position (x, y) is the lower limit on heading ψ reachable with two clothoid segments (the blue curve). Smaller angles can be reached by adding a third clothoid segment to create an S shape as shown in green. Corresponding curvature κ arc length s trace is plotted alongside.

This is because an s-shaped curve cannot be formed with only two segments as illustrated in Fig. 2. In [23] smaller angles are addressed with a second algorithm which computes two matched clothoid pairs (four segments total) instead. The advantage of the method is that the clothoids do not have to be symmetrical, so the final position can be reached with a lower sharpness and peak curvature as well as a shorter path length in some cases compared to using symmetric clothoid pairs.

The condition for the existence of an unsymmetrical clothoid pair to fit three non-collinear points is given by (6) reproduced from [37].

$$\frac{\frac{g}{h} + \cos(\psi)}{\sin(\psi)} < \frac{I_C(\psi)}{I_S(\psi)}. \tag{6}$$

The Fresnel Integrals defining the Cartesian position of a clothoid given in (3) and 4 in Section 2.1 can equivalently be expressed as integrals over deflection δ as defined in (2).

$$x = a \cdot I_C(\delta) = a \cdot \int_0^\delta \frac{\cos u}{\sqrt{u}} du, \tag{7}$$

$$y = a \cdot I_S(\delta) = a \cdot \int_0^\delta \frac{\sin u}{\sqrt{u}} du \tag{8}$$

Equation (6) gives a maximum ratio of g/h in terms of the final angle ψ. The length g is always the distance to the furthest point from the intersection P and h is the distance to the closer point, as shown in Fig. 3. As the condition is given in terms of points rather than poses, the requirement that the points not be collinear is equivalent to (5) because $\psi = \phi$ would indicate collinearity and $\psi < \phi$ would result in the opposite initial heading given by vector $X_1 - P$. In order to avoid very short, high sharpness

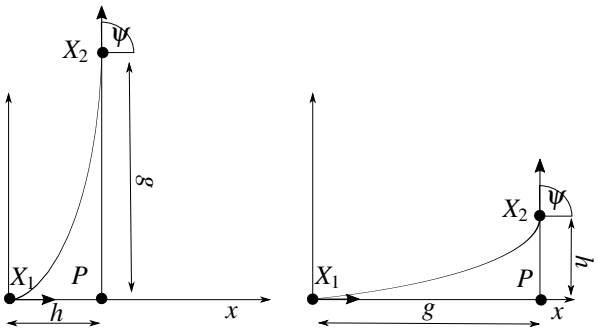


Fig. 3. The two limiting positions for an unsymmetrical clothoid pair without line segments reaching an angle ψ according to the g/h condition in (6). For $\psi = 90$, $\frac{I_C(\psi)}{I_S(\psi)} = 1.7749$.

segments, straight line segments can be introduced before or after the clothoid pair as suggested in [37]. This allows unequal cases beyond the limits shown in Fig. 3 to be fitted.

Using six parameters to describe two matched clothoids with zero curvature at the beginning and end permits the inclusion of a straight line at either end. The parameters would be two for each clothoid segment (total four) and two more for the length of the straight line at either end. The paths created for automated driving by [23] use a matched clothoid pair (two segments) for any corner. With this number of segments, there are certain unreachable poses as an s-shape cannot be formed within one region. The limited flexibility of these paths should be sufficient for executing a turn and returning to zero curvature in each region.

2.3. Obstacle field representation

Any field of polygonal obstacles can be equivalently represented as a set of possibly overlapping convex regions of free space [5]. Path planning within convex regions can be divided into the following steps:

1) Spanning. Convert obstacle representation into a small number of possibly overlapping convex regions which span the free space.
2) Assignment. Assign path segments to a sequence of connected regions between the region containing the start to the region containing the goal
3) Curve Fitting. Solve for the best path from the start to the goal which remains within this sequence over its entire length

The 'Spanning' problem involves calculating a minimum number of spanning regions and is an NP hard problem in itself. A common representation of an obstacle field constructed from range data is an occupancy grid [38]. This consists of a 2D array of cells and can be created from uncertain range measurements from vehicles which are able to estimate their own position [39]. Each cell represents an area of the floor, with a number $p \in [0,1]$ indicating the probability it contains an obstacle. A threshold can be used to create a binary map of occupied and unoccupied cells. The coarse obstacle-free region sets used in the numerical examples can be generated using the vertical decomposition method [40]. This begins with piecewise linear polygons, which can be created by connecting the cell corners of a binary occupancy grid. More complex environments and cluttered obstacle fields could be addressed using Iterative Region Inflation by Semi-definite Programming as described in [41].

The advantage of the regions being convex is that they can be solved guaranteeing optimality (that no other parameter set can minimize the objective further) and completeness (that if a path exists it will certainly be found). Given that the polygonal representation is only an approximation to the real obstacle field, it can only offer completeness within the limits of the resolution used to represent the obstacles but this provides more freedom than a series of pose samples chosen heuristically and may permit lower cost solutions in some cases.

Deits and Tedrake [18] use a mixed integer formulation which addresses 'Assignment' and 'Curve Fitting' simultaneously to achieve optimal results.'Assignment' can also be approached as a graph problem, where there is a node for each region. Edges are inserted between regions which are connected. Connected regions can be found easily in configuration space as the vehicle is reduced to a point and therefore if any corner of a region is contained within any other, it is certain one is reachable from the other and they should be connected in the graph. Using the straight line distance between region corners to give an edge weighting, results in an approximation of the shortest global path. In this way the topology can be solved separately, compensating for one weakness of numerical optimisation, the 'topological blindness' identified in [42].

2.4. Test environment 1

The first test environment was created for a hypothetical item fetch AGV with a payload of 50kg and a top speed of 1m/s.

A simple region shape for exposition is an axis aligned rectangle extending from x_{min} to x_{max} and from y_{min} to y_{max}. Regions for the local avoidance problem are shown in Fig. 4. We assume the assignment is available between N regions and N clothoid pairs. Referring to Fig. 4, the first pair will be assigned to the blue region, the second pair to the red region and the last pair to the amber region.

2.5. Test environment 2

The second test environment shown in Fig. **??** was created for an automated fork lift AGV research platform which is taken to be representative [44]. The obstacles are

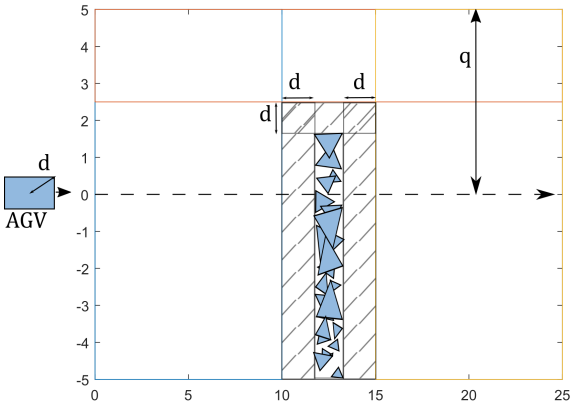


Fig. 4. Example of an obstacle field represented as a set of possibly overlapping empty convex regions. The reference path is a straight dashed line along the x-axis (to the right). The largest dimension of the vehicle body used to expand the obstacles d and the maximum deviation from the path is q.

Table 1. Dimensions from datasheet [43]. *Stopping distance R based on top speed 3.22m/s and hypothetical braking deceleration of 4 m/s^2.

Parameter	Dimension (mm)
W	1067
L	1583
r	1289
L_s	1001
S_0*	1200
R*	1300

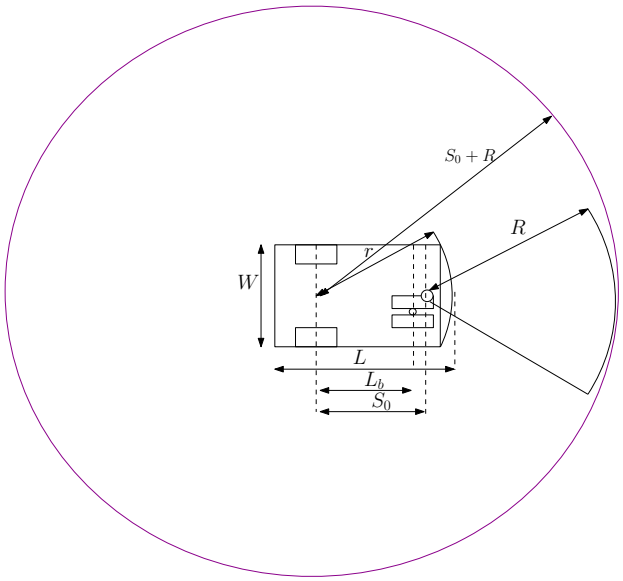


Fig. 5. Dimensions used to expand the obstacles.

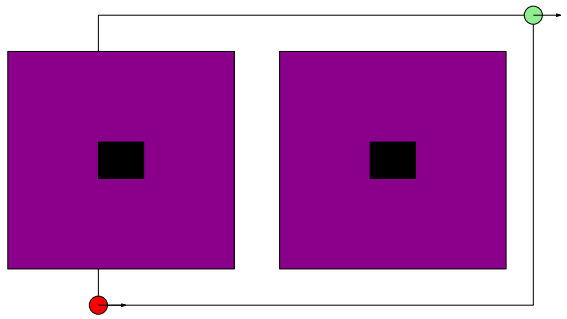


Fig. 6. Pallet environment. Obstacles are black with expansion by the vehicle disk shown in purple.

based on (1.0×1.2)m pallets which are commonly used in the UK and Netherlands [45]. The datasheet for the manually operated vehicle on which the AGV is based, a Hyster E30-40HSD [43] gives the dimensions in Table 1. The plan of the vehicle is given in Fig. 5.

The datasheet gives the maximum speed as 7.2 miles per hour (3.22 m/s). Traction is provided by dual 4.8 kW motors. The unloaded weight of the vehicle is 3059 kg and the battery 1043 kg for a total of 4201 kg [43].

For correct operation it is important to consider the exclusion zone of the safety rated range sensor fitted to the front of the vehicle. If an obstacle breaches the exclusion zone the AGV must perform an emergency stop, or in some cases slow down significantly. To avoid slowing down the path planning must account for not only the shape of the vehicle but also the shape of this zone. Often this is a cuboid slightly wider than the vehicle, sufficiently long that the AGV can come to a complete stop from full speed before the front makes contact with a static obstacle. More details are available in the NIST Safety Standards [46].

In the two simulated environments the obstacle field is represented in 2D. The bounding circle dimension is strongly influenced by the stopping distance R. Starting from the constant acceleration equation $v^2 = u^2 + 2as$ and setting $v = 0$ gives the stopping distance $s = \frac{u^2}{-2a}$.

2.6. Problem definition: Clothoid fitting to convex regions

The core problem is G^2 continuous path planning around obstacles for a point robot. Using a convex spanning region approach described in Section 2.3 this can be divided into a small number of sub-problems. This paper addresses the 'Curve Fitting' sub-problem in detail using the path representation set out in Section 2.2. The approach is described in Section 3.

The requirement for G^2 (curvature) continuity provides four constraints on each segment similar to the posture interpolation problem [9]. In each region we permit one path piece consisting of a line segment length $s_{0,i}$, a matched pair of clothoid segments defined by $\alpha_{1,i}, \alpha_{2,i}, L_{1,i}, L_{2,i}$ and

the other line segment length $s_{F,i}$, as explained in Section 2.2. This provides a total of six free parameters per path piece. As there are only four constraints per region the problem is under-determined and the length of the straight lines can be balanced against the length of the clothoids to find the ideal combination. We consider the start and end region to always be separate even if they overlap, so there will be a minimum of four clothoid segments across the whole spline and s-shapes are feasible.

3. METHOD: CLOTHOID FITTING TO CONVEX REGIONS

The core routine is a method to find the parameters of a sequence of G^2-continuous (continuous in curvature) clothoid segments which is contained entirely within a chain of obstacle-free regions. We minimize the weighted squared sum of path length and sharpness, over a sequence of N path pieces assigned to N convex regions. The objective function is designed to trade off smoothness with path length such that the best path is one that minimizes the weighted squared sum of sharpness and path length to reach the x, y, ψ and κ of the reference path. This is captured in (9) below:

$$\begin{aligned}&\min_{\boldsymbol{\alpha},\boldsymbol{L},\boldsymbol{s_0},\boldsymbol{s_F}} J_T = b\cdot\boldsymbol{\alpha}^T\boldsymbol{\alpha}+\boldsymbol{L}^T\boldsymbol{L}+\boldsymbol{s_0}^T\boldsymbol{s_0}+\boldsymbol{s_F}^T\boldsymbol{s_F}\\ &\text{such that}\begin{cases}\boldsymbol{c_{eq}}(\boldsymbol{\alpha},\boldsymbol{L},\boldsymbol{s_0},\boldsymbol{s_F})=\boldsymbol{0},\\ \boldsymbol{c}(\boldsymbol{\alpha},\boldsymbol{L},\boldsymbol{s_0},\boldsymbol{s_F})\le\boldsymbol{0},\end{cases}\\ &\text{with bounds } \boldsymbol{L}\ge\boldsymbol{0},\boldsymbol{s_0}\ge\boldsymbol{0},\boldsymbol{s_F}\ge\boldsymbol{0}.\end{aligned} \tag{9}$$

Bold variables indicate vectors of the parameters defined in Section 2.1 for every path piece. Scaling parameter b is discussed in Section 3.1. The equality constraints arise from enforcing continuity between each piece and of the first and last segments with the origin and destination respectively. The inequality constraints relate to the requirement to remain outside every obstacle. The first derivatives of the objective and constraints are given in Appendix A.

3.1. Objective function

Two important objectives for an alternative path for an AGV are the total length and the peak sharpness. Both of these properties are desirable rather than mission critical so they are useful in resolving the many solutions which are available to reach a given pose smoothly. Other performance measures such as reaching the destination exactly, limiting the peak curvature and avoiding the obstacles are better interpreted as hard constraints, as it is not useful to compromise them in any way for the measures which are only desirable.

The degree to which smoothness is important compared to path length may depend on the application so it is left with a scaling parameter b. The units of b are m^2 per $\mathrm{radian}^2/\mathrm{m}^4$ = $\mathrm{m}^6\cdot\mathrm{radian}^{-2}$. If b is set very high, smoother paths can be attained, at the cost of increasing path length. In general there is a trade off between path length and smoothness (measured by the least maximum rate of change of curvature as identified by [24]).

Results will be reported with three alternative parameters settings for b, with and without the straight lines, as described below.

- Equal Weighting with Lines, $b=1$, $\min\limits_{\alpha_i,L_i,s_0,s_F} J_T = \sum^N L_i^2+\sum^N \alpha_i^2$
- Equal Weighting no Lines, $b=1$, $s_0=0$, $s_F=0$, $\min\limits_{\alpha_i,L_i} J_T=\sum^N L_i^2+\sum^N \alpha_i^2$
- Minimum Sharpness, no Lines, $b=\inf$, $s_0=0$, $s_F=0$, $\min\limits_{\alpha_i} J_T=\sum^N \alpha_i^2$

3.2. Equality constraints for region i

For continuity, the spline must reach the goal pose and curvature given by $[\hat{x},\hat{y},\hat{\psi},0]^T$. A straight line s_0 before and s_F after the clothoid pair is included while maintaining curvature continuity by the constraint $\kappa=0$ which forces the clothoid pair to be matched, smoothly returning the curvature to zero at the end.

For multiple regions, the continuity constraints are applied implicitly by integrating each segment starting from the final pose of the last. As the curvature is zero at the end of each pair, this can be done by integrating pairwise from the origin as in (10) followed by a rotation and translation to the final pose of the last segment using the $\oplus$ operator detailed in Appendix B. This is an example of single shooting trajectory optimisation as described in [47]. Each region gives rise to one additional $\kappa=0$ constraint, to ensure the straight lines $s_{0,i}$ and $s_{F,i}$ can be added with curvature continuity.

$$\left[\begin{array}{c|c}
x_i & \begin{array}{l} s_{0,i}+C(\alpha_{1,i},L_{1,i},0)\\ \quad+\cos(\delta_{1,i})\cdot C(\alpha_{2,i},L_{2,i},k_{m,i})\\ \quad-\sin(\delta_{1,i})\cdot S(\alpha_{2,i},L_{2,i},k_{m,i})\\ \quad+s_{F,i}\cdot\cos\hat{\psi}_i\end{array}\\ \hline
y_i & \begin{array}{l}\sin(\delta_{1,i})\cdot C(\alpha_{2,i},L_{2,i},k_{m,i})\\ \quad+\cos(\delta_{1,i})\cdot S(\alpha_{2,i},L_{2,i},k_{m,i})\\ \quad+s_{F,i}\cdot\sin\hat{\psi}_i\end{array}\\ \hline
\psi_i & \alpha_{1,i}L_{1,i}^2/2+\alpha_{2,i}L_{2,i}^2/2\\ \hline
\kappa_i & \alpha_{1,i}L_{1,i}+\alpha_{2,i}L_{2,i}
\end{array}\right]. \tag{10}$$

The path in local coordinates for one piece given by (10) can be composed using the $\oplus$ operator to find the final position in global coordinates.

$$\begin{aligned}[X_i,Y_i,\Psi_i]^T=&[X_{i-1},Y_{i-1},\Psi_{i-1}]^T\oplus[x_i,y_i,\psi_i]^T,\\ &\forall\, i\in[1,N],\end{aligned}$$

where capital letters indicate global coordinates. The first region is given the index $i=1$. $[X_0,Y_0,\Psi_0]$ indicates the

starting position. This is taken to be the origin of the coordinate system.

3.3. Equality constraints vector

Subtracting the goal pose from the Nth pose in global coordinates gives the equality constraints, where N is the total number of convex regions under consideration.

$$\boldsymbol{c_{eq}} = \begin{bmatrix} X_N - \hat{X} \\ Y_N - \hat{Y} \\ \Psi_N - \hat{\Psi} \\ \kappa_1 - 0 \\ \vdots \\ \kappa_N - 0 \end{bmatrix} = \mathbf{0}. \tag{11}$$

Note that the $\boldsymbol{c_{eq}}$ array is length $N+3$ as there is a constraint on curvature at the end of each clothoid pair κ_i. The other quantities are scalar and refer to the pose of the final segment at the end of the spline.

3.4. Inequality constraints

In order to operate on the obstacle field directly and avoid spatial sampling, inequality constraints are used to fix the segments inside their assigned regions. The assignment of curve sections to regions will not be discussed here, but can be approached as an integer problem to find the boolean assignment matrix $\boldsymbol{H}^{(N\times R)}$ which results in the lowest cost solution. R is the number of regions in the map, and N is the number of pieces of the spline between the start and the goal. A simple region shape for exposition is an axis aligned rectangle extending from X_{min} to X_{max} and from Y_{min} to Y_{max}. This leads to the following eight inequality constraints on every region, ensuring that both the start and end of each piece are contained.

Based on the assignment $\boldsymbol{H}(i,j)$, indicating path piece i must be entirely contained in region subscript j. The constraints for one region are given by

$$\boldsymbol{H}(i,j) = 1 \iff \boldsymbol{d_i} = \begin{bmatrix} X_i - X_{max,j} \\ -X_i + X_{min,j} \\ Y_i - Y_{max,j} \\ -Y_i + Y_{min,j} \\ X_{i-1} - X_{max,j} \\ -X_{i-1} + X_{min,j} \\ Y_{i-1} - Y_{max,j} \\ -Y_{i-1} + Y_{min,j} \end{bmatrix} \leq \mathbf{0}. \tag{12}$$

These inequality constraints ensure the start and end of the curve piece i assigned to region j remain inside the region. The position at the start of curve i is identical to the position at the end of curve $i-1$ by the application of the continuity constraints. For the first curve the start is fixed at the origin by the choice of coordinate frame. The inequality confirms that the initial position must be in a region of free space for a solution to exist.

The constraints for the entire problem can be constructed by stacking $\mathbf{d_i}$ for each region into a partition vector as follows:

$$\boldsymbol{c_{ineq}} = \begin{bmatrix} \boldsymbol{d_1} \\ \vdots \\ \boldsymbol{d_N} \end{bmatrix} \leq \mathbf{0}. \tag{13}$$

Constraint (12) only applies to the start and end of each clothoid pair, not the entire curve. This works quite well when the curvature remains low but will cause a problem for certain combinations of region shape and path curvature as discussed in Section 4.2.

3.5. Multiple shooting formulation

Alternatively the problem can be posed as a multiple shooting trajectory optimisation by the terminology of [47]. The suggestion is that although there are more numerous parameters and more constraints on the multiple-shooting problem, it may counter-intuitively be easier for the solver because each parameter-constraint pair is more independent and closer to linear. This involves extra parameters $X_{0,i}$, $Y_{0,i}$, $\Psi_{0,i}$ which provide a pose offset for each clothoid pair and explicit continuity constraints between each clothoid pair and the last. The problem would then be described as

$$\begin{aligned} &\min_{\boldsymbol{\alpha},\boldsymbol{L},\boldsymbol{s_0},\boldsymbol{s_F},\boldsymbol{X_0},\boldsymbol{Y_0},\boldsymbol{\Psi_0}} J_T = b\cdot\boldsymbol{\alpha}^T\boldsymbol{\alpha} + \boldsymbol{L}^T\boldsymbol{L} + \boldsymbol{s_0}^T\boldsymbol{s_0} + \boldsymbol{s_F}^T\boldsymbol{s_F} \\ &\text{subject to} \begin{cases} \tilde{\boldsymbol{c}}_{\boldsymbol{eq}}(\boldsymbol{\alpha},\boldsymbol{L},\boldsymbol{s_0},\boldsymbol{s_F},\boldsymbol{X_0},\boldsymbol{Y_0},\boldsymbol{\Psi_0}) = \mathbf{0}, \\ \tilde{\boldsymbol{c}}_{\boldsymbol{ineq}}(\boldsymbol{\alpha},\boldsymbol{L},\boldsymbol{s_0},\boldsymbol{s_F},\boldsymbol{X_0},\boldsymbol{Y_0},\boldsymbol{\Psi_0}) \leq \mathbf{0}, \end{cases} \\ &\text{with bounds } \boldsymbol{L} \geq \mathbf{0},\ \boldsymbol{s_0} \geq \mathbf{0},\ \boldsymbol{s_F} \geq \mathbf{0}. \end{aligned} \tag{14}$$

Now the coordinates are calculated slightly differently, using the new offset parameters rather than a recurrence relation.

$$\begin{aligned} [X_i, Y_i, \Psi_i]^T = &[X_{0,i-1}, Y_{0,i-1}, \Psi_{0,i-1}]^T \oplus [x_i, y_i, \psi_i]^T, \\ &\forall\, i \in [1, N], \end{aligned}$$

where capital letters indicate global coordinates.

Subtracting the final pose and curvature of the previous pair from the initial pose each pair, with zero curvature gives the equality constraints for that region arising from continuity

$$\tilde{\boldsymbol{q}}_{\boldsymbol{i}} = \begin{bmatrix} X_{0,i} - X_{i-1} \\ Y_{0,i} - Y_{i-1} \\ \Psi_{0,i} - \Psi_{i-1} \\ \kappa_{0,i} - 0 \end{bmatrix} = \mathbf{0}. \tag{15}$$

There is also a constraint with the goal pose similar to the single shooting form, which must be included in the stack.

$$\tilde{\boldsymbol{q}}_{\boldsymbol{goal}} = \begin{bmatrix} X_N - \hat{X} \\ Y_N - \hat{Y} \\ \Psi_N - \hat{\Psi} \\ \kappa_N - 0 \end{bmatrix} = \mathbf{0}. \tag{16}$$

Here we use the first subscript of X_0, i to indicate it is the X offset for the region given by the second subscript. The offsets are new search parameters for this formulation. Symbols with a single subscript, the subscript identifies the region and X_i refers to that at the end of the curve assigned to that region. Again the first region is given the index $i = 1$. The start pose is $[X_0, Y_0, \Psi_0]$. This is taken to be the origin of the coordinate system.

The inequality constraints for the entire problem can be constructed by stacking $\tilde{k}_i$ for each region into a partition vector as follows:

$$\tilde{\boldsymbol{c}}_{\boldsymbol{ineq}} = \begin{bmatrix} \tilde{\boldsymbol{q}}_{\boldsymbol{1}} \\ \vdots \\ \tilde{\boldsymbol{q}}_{\boldsymbol{N}} \\ \tilde{\boldsymbol{q}}_{\boldsymbol{goal}} \end{bmatrix} = \boldsymbol{0}. \tag{17}$$

The inequality constraints for region i are given by

$$\boldsymbol{H}(i,j) = 1 \iff \tilde{\boldsymbol{d}}_{\boldsymbol{i}} = \begin{bmatrix} X_i - X_{max,j} \\ -X_i + X_{min,j} \\ Y_i - Y_{max,j} \\ -Y_i + Y_{min,j} \\ X_{i-1} - X_{max,j} \\ -X_{i-1} + X_{min,j} \\ Y_{i-1} - Y_{max,j} \\ -Y_{i-1} + Y_{min,j} \end{bmatrix} \leq \boldsymbol{0}. \tag{18}$$

The constraints for the entire problem can be constructed by stacking $\tilde{d}_i$ for each region into a partition vector as follows:

$$\tilde{\boldsymbol{c}}_{\boldsymbol{ineq}} = \begin{bmatrix} \tilde{\boldsymbol{d}}_{\boldsymbol{1}} \\ \vdots \\ \tilde{\boldsymbol{d}}_{\boldsymbol{N}} \end{bmatrix} \leq \boldsymbol{0}. \tag{19}$$

4. NUMERICAL RESULTS

First, in Section 4.1 the suitability for finding the smoothest path subject to obstacle constraints is tested on an environment similar to Fig. 4. Subsequently, an alternative formulation, the effect of analytical gradients, tuning parameter b and the objective function are evaluated for their potential in speeding up the solution. A number of tests without obstacles are included in Section 4.5 and 4.6 so path parameters can be compared with an existing heuristic method.

4.1. Motivating problem

The tested problem concerns the avoidance of a small obstacle blocking a straight path while remaining within a set tolerance from it. The allowable distance from the original path is used to generate the outer boundary enclosing [0, 5] and [11, -5]. With the obstacle information available from sensors this is broken up into three convex regions. As set out in Section 2.2 this necessitates a path with six

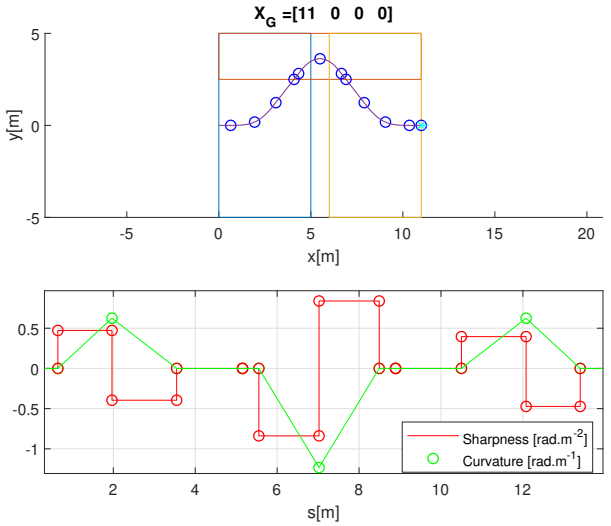


Fig. 7. Single shooting optimal avoidance path for a vehicle traveling along the x-axis which encounters a rectangular obstacle at [5,-2.5], extending to [6,2.5], rejoining the reference path at [11, 0]. Weighting parameter $b = 1$.

clothoid segments, two at the start, two at the goal and two at an intermediate region needed to connect the two.

The single shooting formulation was solved using *interior-point*[1] method in 36 iterations to produce the path and curvature profile shown in Fig. 7. The region constraints are satisfied. This can be seen as each of the four segments is marked with an open circle. The same points are marked with open circles in the curvature and sharpness profile shown below the x-y plot in Fig. 7. These can be used to evaluate the quality of the path without reference to a specific vehicle model. The largest magnitude curvature is found at the midpoint with a value of -1.233 [m^{-1}], corresponding to a turning radius of 811mm, suitable for a small vehicle with Ackermann steering, having a wheelbase less than 811mm. The largest magnitude sharpness of 0.8388 [$radm^{-2}$] is seen on the segments before and after the peak curvature. This path can be tracked with high accuracy by different AGVs by reducing forward speed to control the lateral acceleration based on the path curvature, and the angular acceleration based on the sharpness.

The single shooting formulation reduces the number of parameters which might be expected to reduce total execution time. In fact, it took multiple seconds to reach convergence on the small 20m$\times$10m environment made up of three regions tested.

4.2. Weighting parameter b effects

The weighting parameter b has a significant effect on convergence time as shown in Fig. 8. The number of iterations and function evaluations increased for larger weights, in the same way as time to convergence.

The two path plots cover both extremes of weighting tested for the avoidance problem. Fig. 9(b) shows the effect of a small weighting on sharpness. The path is shorter, the peak sharpness is higher, but it is not the shortest continuous path which would pass through the corner formed by region one and two (the top one). The segments are close to equal length because, for a set of numbers with a fixed sum, the sum of squares will be minimized if the numbers are equal. This provides a bias toward equal length segments and may contribute to the success of an equal weighting of the two components.

Fig. 9(a) shows the effect of a large weighting on sharpness. The path is longer and more meandering but has lower peak sharpness. The constraints are met as the start and end are contained but the curve leaves the convex region at the top. Because the region constraint is not applied to the point of peak curvature, there are a large number of feasible solutions with similar sharpness. This makes an objective based on sharpness ($b >> 1$) very flat close to the minimum with this set of obstacles. Looking through the text output of *fmincon*, a feasible solution according to the threshold of 1e-6 is found earlier but the search continues until the threshold is reached. To solve this problem, the region constraints need to be be applied to additional samples along the path.

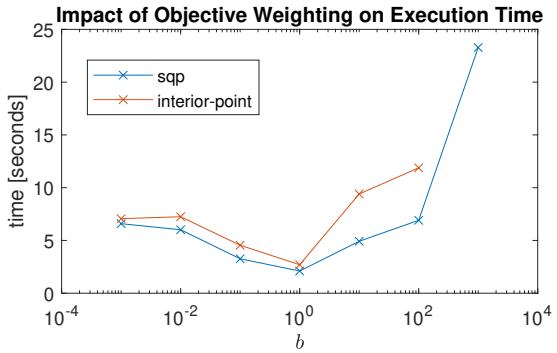
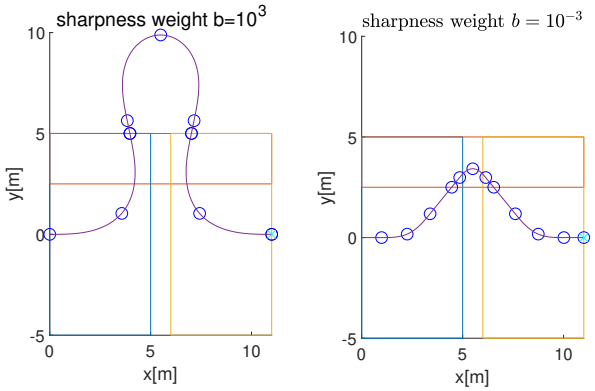


Fig. 8. Convergence time as weighting b between sharpness and length is varied.



(a) $b = 10^3$. (b) $b = 10^{-3}$.

Fig. 9. Path comparison for extremes of b.

4.3. Obstacle avoidance multiple shooting formulation

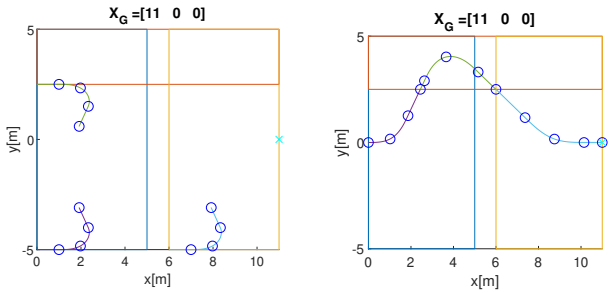
In the multiple shooting formulation by contrast, the curve positions in one region are independent of the parameters of the earlier segments due to the introduction of a new $[X_0, Y_0, \Psi_0]$ offset parameter for each region and a new constraint that this offset matches the final pose of the curve in the preceding region.

With the initial guess shown in Fig. 10(a) the solver must find offsets which satisfy both continuity and region containment. This can give better performance than single shooting in some cases involving longer splines [47]. In the problem shown in Fig. 10(b) with three regions the discontinuities have been resolved after convergence, the execution time is comparable with single shooting, slightly improved, with around 700 function calls using interior point method.

This is despite increasing the number of parameters from six per region to nine per region (18 to 27 with $N = 3$), and the number of constraints from $4+N$ equality and $8N$ inequality (7 and 24 with $N = 3$) to $4+4N$ equality and $8N$ inequality (16 and 24 with $N = 3$). The number of inequality constraints could be reduced slightly for the multiple shooting by fixing the start position at the world origin, to match the features of the single shooting setup.

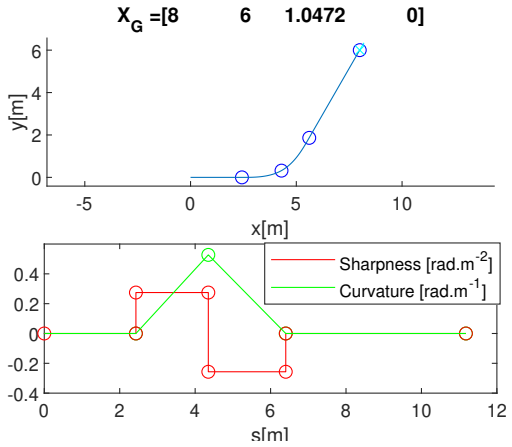
4.4. Curve fitting with two clothoids with different objectives

Cartesian path and curvature profile results for a single region containing two clothoid segments and two straight lines are shown in Fig. 11(a) for comparison to those reported by [23] for two matched clothoids. For illustration the parameters identified for the same final pose by our

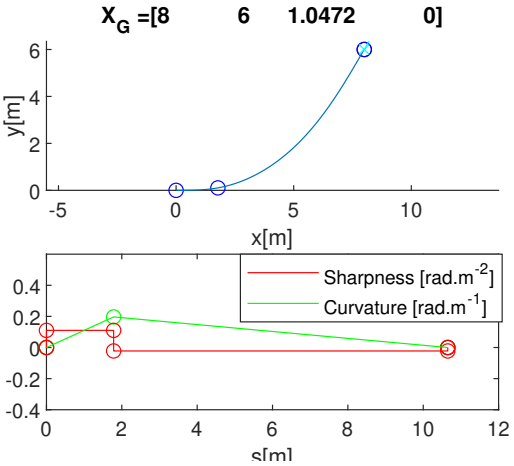


(a) Initial parameter guess. (b) After convergence.

Fig. 10. Multiple shooting initial guess ($\alpha_i = 1.0$, $L_i = 1.0$) showing path continuity constraints are not met until the optimisation has converged.



(a) Equal Weighting with Lines.



(b) Minimum Sharpness, no Lines.

Fig. 11. Side by side comparison of a path from the origin to [8,6, 60], curvature profile is shown below.

Table 2. Parameters identified for a path from [0,0,0] to [8,6,60]. The objectives are explained in Section 3.1.

	Equal Weighting, with Lines	Minimum Sharpness, no Lines
α_1 [m^{-2}]	0.1472	0.1094
α_2 [m^{-2}]	-0.1135	-0.0222
L_1 [m]	2.4893	1.7981
L_2[m]	3.2281	8.8535
s_0 [m]	1.5966	0
s_F [m]	3.7390	0
Total Length [m]	11.0529	10.6516
κ_m [m^{-1}]	0.3633	0.1966
interior-point iterations	12	2000+
interior-point funcCount	101	16220+
sqp iterations	15	12
sqp funcCount	130	89

method with two different cost functions are shown in Table 2.

There are two remarks to make here. Firstly, the [8,6,60] point meets the condition in (6), Section 2.2 and can be reached exactly with two clothoids and no line segments. Many nearby points require a line to be included at the end of the clothoid pair for convergence, particularly those with lower final angles (closer to the limit given by (5)). Secondly, the initial guess had to be very close to the minimum in order to reach convergence at all with *interior-point* method for this problem. Using the 'Equal Weighting with Lines' objective an initial guess of [1, -1, 1, 1, 1, 1] converged in 12 iterations with *interior-point* method. With the same initialisation vector and the 'Minimum Sharpness, no Lines' objective, *interior-point* did not converge within 2000 iterations. If the solver was changed to *sqp* convergence took 12 iterations and 89 function evaluations. The *interior-point* method is shown to be slightly faster on some problems but less stable than *sqp* on the limited numerical tests performed. These tests only involved three regions, leading to 18 parameters and 16 constraints. It is a notable advantage of the convex region representation that the number of regions can be strictly limited even in environments with lots of clutter. The documentation lists *sqp* as a medium-scale algorithm, which needs to store and operate on matrices with the dimension of the parameters [48]. If problems are encountered in larger tests, *interior-point* is a large scale-algorithm which does not rely on dense matrix operations, and should perform better. Due to the general constrained form of the problem other highly optimized methods for specific performance needs could be used depending on the needs of the application (e.g., limited processing time, limited memory, embedded platform...).

The 'Minimum Sharpness, no Lines' solution would be expected to be longer with lower peak sharpness than the 'Equal Weighting, with Lines' solution, as only reducing sharpness contributes to the objective. As expected Table 2 shows the peak sharpness is reduced from 0.1472 to 0.1094, around 30%. Surprisingly the total length of the 'Minimum Sharpness, no Lines' solution is also less than the 'Equal Weighting, with Lines'. In this case *sqp* is stuck in a local minimum when additional straight lines are included. Line segments must be permitted for convergence on cases outside the boundary of the existence criteria in Section 2.2, but it seems that well within the boundary such as the end point in Fig. 11(a), considering line segments may lead to the introduction of local minima.

4.5. Curve fitting with two clothoids with analytical gradient

The impact of analytical gradients was tested on a single region containing two segments to the point [8, 6, 40], close to the lower angle limit, as shown in Fig. 12(b). With an initial guess which did not meet the regions constraints

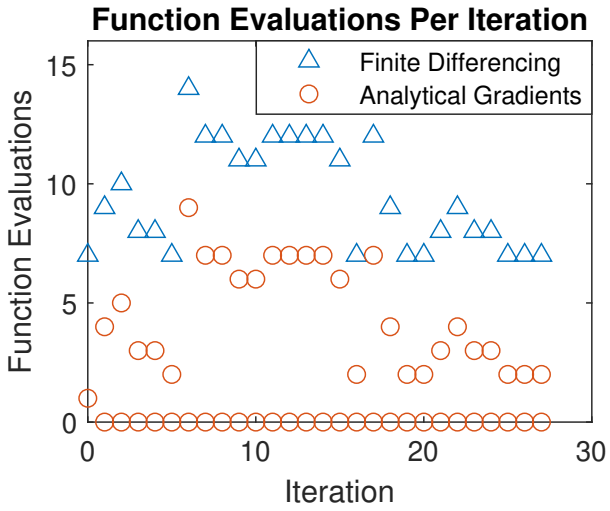
$\boldsymbol{p} = [0.1, -0.1, 1, 1, 1, 1]$, *interior-point* method[1] failed to converge. The alternative algorithm *sqp*[1] converged in 27 iterations with or without derivatives as shown in Fig. 12(a). When using analytical derivatives the number of function evaluations was reduced from 264 total to 123 total but the total execution time increased from 1.33 seconds to 2.28 seconds[2]. Therefore the mean execution time per function evaluation increased from $1.33/264 = 5$ ms to $2.28/123 = 18$ ms with the additional of numerical gradients.

On a similar problem with a different goal heading, providing analytical derivatives reduced the function count reported by *fmincon* from 213 to 55 and the number of iterations to reach the same optimality threshold from 25 to 24. However the computation time increased by about 50% from 2.35 seconds to 3.00 seconds. If the solver was changed to *sqp*[2], the effect was similar. Slightly reduced count of function evaluations but an increased computation time. This may be due to the numerous integral terms in the expressions for the gradients taking more time to evaluate than the function at multiple locations to allow differencing. There are 22 integrals to evaluate in the Jacobian. Only two integrals are required to evaluate the objective function. Equations (3)-(4) were evaluated with Vectorized Adaptive Quadrature method [49] with a relative tolerance[3] of 10^{-6} . No attempt was made to store and reuse repeated terms in the Jacobian, although there are several.

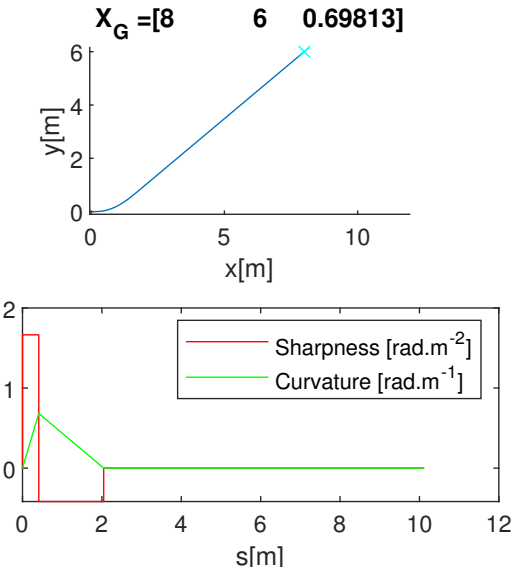
4.6. Curve fitting with four clothoids with different objectives

In order to enable further comparison with the bisection method of Gim *et al*. [23], another test was reproduced involving an s-shaped path comprising four clothoids to the pose $[12, 10, \theta]$, where θ varied in increments of 10 degrees. This shows clearly the trade off between path length and peak sharpness.

The 'Equal Weighting, with Lines' set of paths is shown in Fig. 13. For the final angle of -30 degrees the key parameters of the solution are shown in Table 3. The cost is equivalent to penalizing path length with $b = 1$ [m^6 rad^{-2}] but because each segment is squared individually, the cost is lower if each segment is of similar length. The 'Minimum Sharpness, no Lines' set of paths for comparison is shown in Fig. 14. For every tested heading the path is significantly longer than the 'Equal Weighting, with Lines' solution, while the peak sharpness is much lower. This makes intuitive sense because over longer distances the shortest path is a straight line. As the sharpness is increased the path comes to resemble the point to point line more closely.



(a) Both took 27 iterations so are shown on the same figure.



(b) Both produced exactly the same path and curvature profile.

Fig. 12. Comparison of performance with and without analytical gradients. The number of iterations is high because the heading angle at the end is close to the lower limit given by (5) for this position.

Convergence with the 'Equal Weighting, with Lines' cost was particularly strong in the experiments attempted compared to either the more natural absolute sum of the segment lengths, all squared $J = (\sum_j |\alpha_j|)^2 + (\sum_j |L_j|)^2 + (\sum_j |s_{0,j}|)^2 + (\sum_j |s_{F,j}|)^2$ or the minimum sharpness with the line segments fixed at zero so $J = (\sum_j |\alpha_j|)^2$. Again the minimum sharpness approach produced paths almost identical to the bisection method proposed by [23] on the examples they reported.

[1] *interior-point* and *sqp* are available as options for the *fmincon* function of MATAB.

[2] Numerical tests ran on a consumer laptop with 16GB RAM and an Intel(R) Core(TM) i5-8250U CPU @1.60GHz.

[3] Vectorized Adaptive Quadrature is available in MATLAB as a built in function *q=integral(fun, xmin, xmax)*, the default upper bound on error is $10^{-6} \times q$.

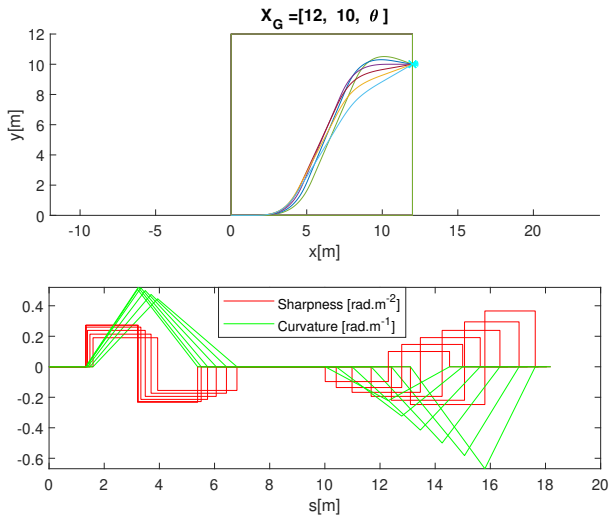


Fig. 13. Six lane change paths with objective 'Equal Weighting, with Lines', each ending at the same point [12,10] with a heading θ separated by 10 degrees.

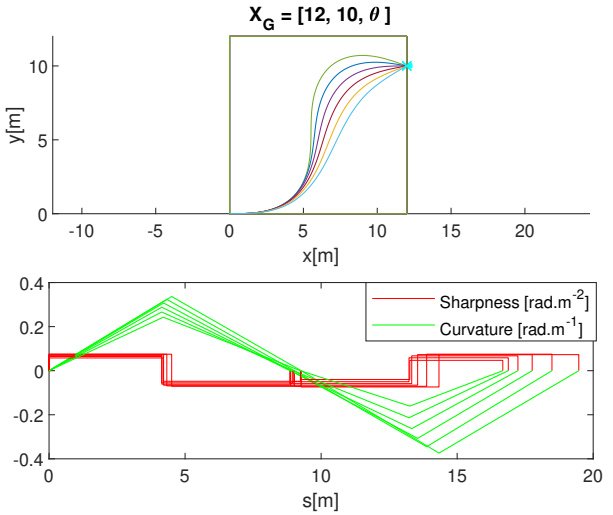


Fig. 14. Six lane change paths with objective 'Minimum sharpness, no lines', each ending at the same point [12,10] with a heading θ separated by 10 degrees.

Table 3. Parameters identified for a path from [0,0,0] to [12,10,-30]. This is one of the curves plotted in Fig. 13. The objectives are explained in Section 3.1 Units α m^{-2}, κ m^{-1}, L m.

	Equal Weighting with Lines	Equal Weighting, no Lines	Minimum Sharpness, no Lines
α_1	0.1707	0.0552	0.0727
α_2	-0.1404	-0.0443	-0.0709
α_3	-0.2808	-0.0894	-0.0745
α_4	0.4582	0.1716	0.0665
Peak Sharpness $\max\|\alpha_i\|$	0.4582	0.1716	0.0745
L_1	2.4569	4.8616	4.7639
L_2	2.9860	6.0572	4.8878
L_3	2.7117	5.4077	5.2704
L_4	1.6615	2.8180	5.967
s_{01}	1.7170	0	0
s_{F1}	3.2565	0	0
s_{02}	3.2565	0	0
s_{F2}	0.0893	0	0
Total Length	18.14	19.1445	20.8288
κ_{m1}	0.4142	0.2685	0.3464
κ_{m2}	-0.7557	-0.4837	-0.3928

The final position was taken from [23] so it can be used to compare the curvature profile and path trace to the one produced by the bisection method presented in that paper. This is a root finding approach which is suitable for meeting the constraint on final position. No objective function is defined so the first solution which meets the constraint tolerance will be accepted. The 'Minimum sharpness, no Lines' path trace in Fig. 14 looks very similar to the paths produced by the bisection method. The total length of both is around 20.8m. The coordinate system differences make side by side comparison a little challenging, as they used a start position of $[0,0,90]$ ending at $[10,12,120]$ which is the same path subject to some affine transformations. The curvature profile is comparable side-by-side and shows the sharpness of each section found by bisection to be close to 0.07125, very close to the average of the 'Minimum Sharpness, no Lines' column in Table 3.

The optimisation method has a clear advantage as it searches intermediate angles by continuously varying the length and sharpness of the first two clothoid segments. Bisection with four clothoids by contrast only searches very coarsely over intermediate angles (every 10 degrees). Searching more effectively should be a big advantage of using optimisation, but in this particular example the improvement is very small. The main benefit of the new method is the ability to take into account obstacle constraints at the start and the end and find a path purely from a polygonal representation of the obstacles.

A range of curves with θ varying from -30 degrees to 20 degrees is shown in Fig. 13. The parameters can be compared to the same range of angles with a cost function which only penalizes the sum of squared sharpness of each segment and forces the straight lines at the start and end of each segment to zero in Table 3. The sharpness is reduced but the total path length is increased until it is almost identical to the curve plotted in [23]. The added value of using four segments over three is questionable as the middle two take almost the same value.

The expected trade off between peak sharpness and to-

tal path length can be seen in Table 3.

4.7. Effect of small changes to region boundaries

When the convex regions are constructed based on an occupancy grid as suggested in Section 2.3 measurement errors may lead to sudden changes in the size of the free space as the probability of one cell being occupied crosses the threshold. In this situation the size of the position shift would be determined by the cell size of the binary occupancy grid. The cell size is typically made larger than the sensor noise by some constant factor. A cell size of 100mm was selected to provide an ample boundary for a common sensor with zero mean error, 10mm standard deviation [50].

The offset error was varied in increments of 100mm, and the resulting change in objective function is shown in Fig. 15. Both curve types increase linearly with the position error at first. The cubic objective is the second derivative of position, while the clothoid minimized 'Equal Weighting With Lines' from Section 3.1 The clothoid curve objective increases more slowly until the final offset of 0.5m which causes a step change in the objective. This seems to be related to the lack of a constraint on the midpoint of the curve piece. As the obstacle is shifted the path must deflect more and in this case the midpoint leaves its assigned region. The problem of sensor errors pushing the path planner into a suboptimal local minimum could be significant. An additional constraint the midpoint should resolve it in this case, but for real world implementation some kind of safety checks will be needed before attempting to drive a new path for situations like this.

4.8. Curvature comparison with cubic spline on environment 2

Side by side comparison reveals the clothoid curve in Fig. 16 has a higher peak curvature and is much smoother as measured by the peak sharpness. Cubic curves are simple to implement and often used for path planning. The cubic curve only has enough degrees of freedom to meet the heading constraint at the start and end by relaxing the constraint on second derivative to zero at the start and the end. The cubic curve in Fig. 17 is continuous and the peak of 1.5m is feasible with a limited steering angle vehicle, however the curvature plot reveals rapid changes in curvature. The sharpness peaks at the start and end and is too large to be shown on the figure. This is will result in the steer-drive wheel actuating rapidly and likely greater tracking error by an AGV attempting to follow it. Using a higher order polynomial could improve on this, but not to the extent of the clothoid curve which has piecewise constant sharpness and the magnitude is always less than $2[\mathrm{m}^{-2}]$ so it can be tracked by a real AGV such as the one

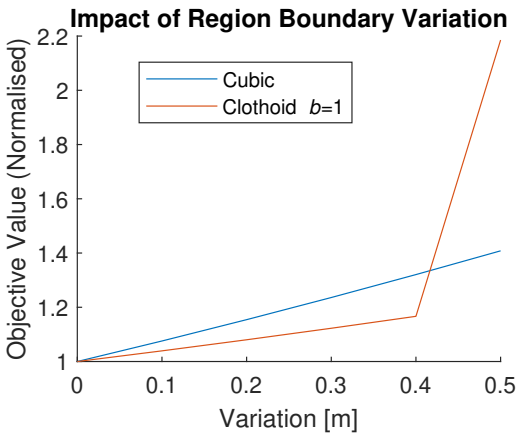


Fig. 15. Normalised objective function against a linear shift of the boundary in x and y.

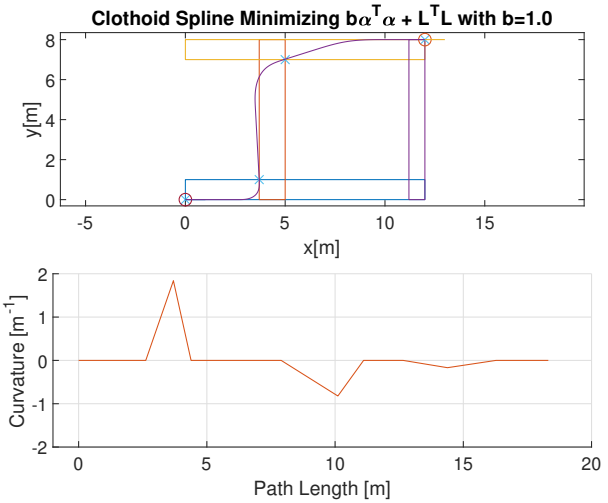


Fig. 16. Path trace and curvature for clothoid spline.

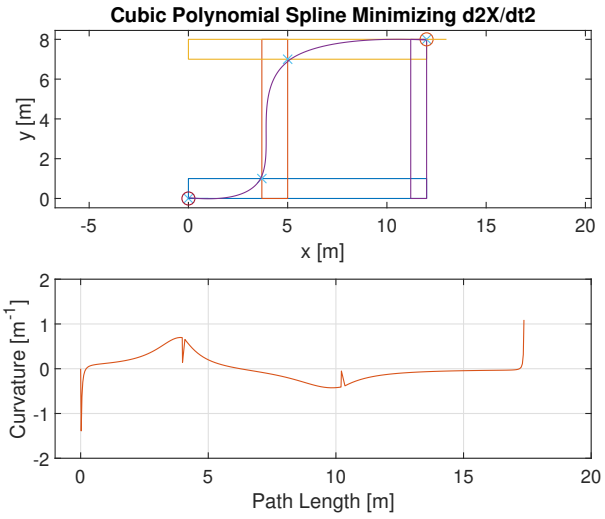


Fig. 17. Path trace and curvature for cubic polynomial spline.

described in Section 2.5.

5. CONCLUSION

The formulation presented in Section 3 is sufficient to solve the path planning with obstacle constraints using clothoid curve segments with G^2 continuity, using a generic constrained non-linear solver. The path with three regions converges reliably and it can be extended to more complex scenes. Execution time of several seconds indicates a need for further optimisation, but the path is equally smooth as the interpolation method of Gim et al [23], without the need for a sequence of points from manual driving or a heuristic for selection of waypoints based on the obstacle field, likely to be suboptimal.

The key concern of industrial automation which it addresses is optimal planning with clothoid curves based on a polygonal obstacle representation which can be constructed from range sensors such as LIDAR. The method is therefore most useful for applications where path quality is important and updates are required less frequently. For example, a centralized adaptive replanning module which is tightly integrated with a central routing algorithm which collects information from a number of sensors and vehicles and makes safety critical path alteration decisions as discussed in [1]. Interaction with other vehicles may be based on the allocation of regions at different times. Validating the interactions becomes much easier if the information is collected from different sources (around blind corners for example) and a safe alternative path is generated which can be followed by the existing vehicle control system.

Freedom to choose a different objective function or weighting parameter b to generate different shape paths is a strength of this formulation but also reveals the weakness of using an optimisation approach. Different objective functions have different convergence characteristics for the same boundary conditions. One based on an equal weighting converges nicely but as b is increased, emphasising smoothness convergence takes longer as the constraints are only enforced at the start and end of each segment. The search of infeasible solutions leads to wasted iterations, especially when $b > 100$. The appropriate weighting for b might best be addressed using knowledge of the speed controllers to be used for path traversal in order to plan minimum time paths around obstructions and minimize the delay from obstructions in a site.

Real-time performance is important for the method to be useful as part of a AGV control system. The multiple shooting form was not much faster on the small environments tested. The analytical Jacobian of the constraints was effective at reducing the number of function evaluations to reach convergence but not total execution time. An improved implementation with more reuse of terms could approach the factor of two reduction in function evaluations recorded. Even with this improvement, the presented method would be slower than existing heuristic methods for clothoid parameter estimation. Its strength lies in the guarantee of optimality given the obstacle polygons (subject to convergence, which was strong in the limited numerical examples tested). Other avenues to improve performance in further work should explore improving the initial parameter guess based on the point to point solution between the corners of the regions, and the use of alternative numerical methods designed for convex constrained problems such as CVX [51]. Finally, even with execution time of several seconds it is possible to use automatic adaptive paths in some situations, as the clothoid shape allows reuse of the same optimal path by vehicles with differing dynamics by adjustment of the longitudinal speed to control the steering rate.

Expanding the environment with a disc the size of the largest dimension of the vehicle is a gross approximation, which will lead to over cautious paths. It would be better to represent the obstacles in (2 + 1) dimensions, and plan in (x, y, ψ). Obstacles can then be expanded based on the current heading of the path as described in [18] who use a sum of squares approach to find (2+1)D convex polygons. Applying these (2+1)D constraints to the optimisation presented here would be straightforward but the number of path pieces needed to make the entire space reachable may be different. This could be investigated in a future study.

Testing a wider range of environments is left to further work. This may motivate an increase in the number of obstacle constraint points on the path to prevent clipping problems seen in Section 4.2. A constraint on the point of peak curvature may be most helpful before sampling the curve at approximately linear intervals. We were not able to prove the statement “provided N connected regions can be found between two poses, there always exists a smooth path between them comprised of line segments and clothoids in the order L-C-C-L”. A wider range of test cases would give confidence that this is true in many cases. Further testing would also be useful to identify cases where a path exists but the peak curvature is so high a vehicle would have to slow down unacceptably to traverse it.

APPENDIX A: ANALYTICAL GRADIENTS OF THE SINGLE SHOOTING FORMULATION WITHIN ONE REGION

Stacking the parameters for the optimisation described in Section 3 into a single vector $\boldsymbol{p}^{1\times 6} = [\alpha_1, \alpha_2, L_1, L_2, s_0, s_F]^T$ allows the Jacobian of the objective for one region to be expressed as

$$\frac{\partial \boldsymbol{J}}{\partial \boldsymbol{p}} = 2 \cdot [b\alpha_1, b\alpha_2, L_1, L_2, s_0, s_F]^T . \tag{A.1}$$

The pose at the end of the path piece in the region of interest is $[X_F, Y_F, \Psi_F]$. For the derivatives we only consider one path piece, and drop the i subscript used in (10). The equality constraint vector with goal pose $[\hat{X}, \hat{Y}, \hat{\Psi}]$ is given by

$$\boldsymbol{c_{eq}} = \begin{bmatrix} X_F - \hat{X} \\ Y_F - \hat{Y} \\ \Psi_F - \hat{\Psi} \\ \kappa_F - 0 \end{bmatrix} = \mathbf{0}, \tag{A.2}$$

and the inequality constraints to remain within a square region bounded by $[X_{max}, X_{min}, Y_{max}, Y_{min}]$ is given by

$$\boldsymbol{c_{ineq}} = \begin{bmatrix} X_F - X_{max} \\ -X_F + X_{min} \\ Y_F - Y_{max} \\ -Y_F + Y_{min} \end{bmatrix} \leq \mathbf{0}. \tag{A.3}$$

The matrix derivatives $\frac{\partial \boldsymbol{c}_{eq}}{\partial \boldsymbol{p}}^{(6\times4)}$ and $\frac{\partial \boldsymbol{c}_{ineq}}{\partial \boldsymbol{p}}^{(6\times4)}$ were constructed from scalar partial derivatives listed below. Each element is defined in terms of the parameters $\boldsymbol{p}$ in the following sections. The symbols are introduced in Section 2.1, the starting pose is taken to be the origin, so the end pose in global coordinates is given by (10). Functions $C(\alpha, s, \kappa)$ and $S(\alpha, s, \kappa)$ are defined by (3) and (4), respectively. The peak curvature where the two clothoids meet is given the symbol $\kappa_m = \alpha_1 L_1$ and the final heading at the end of the second clothoid is given by $\Psi_F = \frac{\alpha_1 L_1^2}{2} + \frac{\alpha_2 L_2^2}{2}$.

A.1. Components with respect to α_1

$$\begin{aligned}
&\frac{\partial X_F}{\partial \alpha_1} \\
&= \int_0^{L_1} -\frac{u^2}{2} \sin\left(\frac{\alpha_1 u^2}{2}\right) du \\
&\quad + \cos(\delta_1) \int_0^{L_2} (-L_1 \cdot u) \sin\left(\alpha_1 L_1 \cdot u + \frac{\alpha_2 u^2}{2}\right) du \\
&\quad + (-\sin(\delta_1) L_1^2/2) \int_0^{L_2} \cos(\alpha_1 L_1 + \alpha_2 u) du \\
&\quad - \sin(\delta_1) \int_0^{L_2} (L_1 \cdot u) \cos\left(\alpha_1 L_1 \cdot u + \frac{\alpha_2 u^2}{2}\right) du \\
&\quad - \left(\cos(\delta_1) L_1^2/2\right) \int_0^{L_2} \sin(\alpha_1 L_1 + \alpha_2 u) du \\
&\quad + s_F (L_1 L_2 + L_1^2/2)(-\sin\Psi_F),
\end{aligned} \tag{A.4}$$

$$\begin{aligned}
&\frac{\partial Y_F}{\partial \alpha_1} \\
&= \int_0^{L_1} \frac{u^2}{2} \cos\left(\frac{\alpha_1 u^2}{2}\right) du \\
&\quad + \sin(\delta_1) \int_0^{L_2} (-L_1 \cdot u) \sin\left(\alpha_1 L_1 u + \frac{\alpha_2 u^2}{2}\right) du \\
&\quad + (\cos(\delta_1) \frac{L_1^2}{2}) \int_0^{L_2} \cos\left(\alpha_1 L_1 u + \frac{\alpha_2 u^2}{2}\right) du \\
&\quad + \cos(\delta_1) \int_0^{L_2} (L_1 \cdot u) \cos\left(\alpha_1 L_1 u + \frac{\alpha_2 u^2}{2}\right) du \\
&\quad + (-\sin(\delta_1) \frac{L_1^2}{2}) \int_0^{L_2} \sin\left(\alpha_1 L_1 u + \frac{\alpha_2 u^2}{2}\right) du \\
&\quad + s_F (L_1 L_2 + L_1^2/2) \cos(\Psi_F),
\end{aligned} \tag{A.5}$$

$$\partial \Psi_F / \partial \alpha_1 = \frac{1}{2} L_1^2 + L_1 L_2, \tag{A.6}$$

$$\partial \kappa_F / \partial \alpha_1 = L_1. \tag{A.7}$$

A.2. Components with respect to α_2

$$\begin{aligned}
\frac{\partial X_F}{\partial \alpha_2} &= \cos(\delta_1) \int_0^{L_2} \left(\frac{-u^2}{2}\right) \sin\left(\alpha_1 L_1 u + \frac{\alpha_2 u^2}{2}\right) du \\
&\quad - \sin(\delta_1) \int_0^{L_2} \left(\frac{u^2}{2}\right) \cos\left(\alpha_1 L_1 u + \frac{\alpha_2 u^2}{2}\right) du \\
&\quad + s_F (-\sin\Psi_F) \left(\frac{L_2^2}{2}\right),
\end{aligned} \tag{A.8}$$

$$\begin{aligned}
\frac{\partial Y_F}{\partial \alpha_2} &= \sin(\delta_1) \int_0^{L_2} -\frac{u^2}{2} \sin\left(\alpha_1 L_1 u + \frac{\alpha_2 u^2}{2}\right) du \\
&\quad + \cos(\delta_1) \int_0^{L_2} \frac{u^2}{2} \cos\left(\alpha_1 L_1 u + \frac{\alpha_2 u^2}{2}\right) du \\
&\quad + s_F (\cos\Psi_F) \cdot \left(\frac{1}{2} L_2^2\right),
\end{aligned} \tag{A.9}$$

$$\frac{\partial \Psi_F}{\partial \alpha_2} = \frac{1}{2} L_2^2, \tag{A.10}$$

$$\frac{\partial \kappa_F}{\partial \alpha_2} = L_2. \tag{A.11}$$

A.3. Components with respect to L_1

$$\begin{aligned}
&\frac{\partial X_F}{\partial L_1} \\
&= \cos(\alpha_1 L_1^2/2) \\
&\quad + \cos(\delta_1) \int_0^{L_2} (-\alpha_1 \cdot u) \sin\left(\alpha_1 L_1 u + \frac{\alpha_2 u^2}{2}\right) du \\
&\quad + (-\sin(\delta_1) \alpha_1 L_1) \int_0^{L_2} \cos\left(\alpha_1 L_1 u + \frac{\alpha_2 u^2}{2}\right) du \\
&\quad - \sin(\delta_1) \int_0^{L_2} (\alpha_1 \cdot u) \cos\left(\alpha_1 L_1 u + \frac{\alpha_2 u^2}{2}\right) du \\
&\quad - (\cos(\delta_1) \alpha_1 L_1) \int_0^{L_2} \sin\left(\alpha_1 L_1 u + \frac{\alpha_2 u^2}{2}\right) du \\
&\quad - s_F (\alpha_1 L_1 + \alpha_1 L_2) \sin(\Psi_F),
\end{aligned} \tag{A.12}$$

$$\begin{aligned}
&\frac{\partial Y_F}{\partial L_1} \\
&= \sin(\alpha_1 L_1^2/2) \\
&\quad + \sin(\delta_1) \int_0^{L_2} (-\alpha_1 \cdot u) \sin\left(\alpha_1 L_1 u + \frac{\alpha_2 u^2}{2}\right) du \\
&\quad + (\cos(\delta_1) \alpha_1 L_1) \int_0^{L_2} \cos\left(\alpha_1 L_1 u + \frac{\alpha_2 u^2}{2}\right) du
\end{aligned}$$

$$+\cos(\delta_1)\int_0^{L_2}(\alpha_1\cdot u)\cos\left(\alpha_1 L_1 u+\frac{\alpha u^2}{2}\right)du$$
$$+(\sin(\delta_1)\alpha_1 L_1)\int_0^{L_2}\sin\left(\alpha_1 L_1 u+\frac{\alpha_2 u^2}{2}\right)du$$
$$+s_F(\alpha_1 L_1+\alpha_1 L_2)\cos(\Psi_F), \tag{A.13}$$
$$\frac{\partial\Psi_F}{\partial L_1}=\alpha_1 L_1+\alpha_1 L_2, \tag{A.14}$$
$$\frac{\kappa_F}{\partial L_1}=\alpha_1. \tag{A.15}$$

A.4. Components with respect to L_2

$$\frac{\partial X_F}{\partial L_2}=\cos\left(\alpha_1 L_1 L_2+\frac{\alpha_2 L_2^2}{2}\right)\cos(\delta_1)$$
$$-\sin\left(\alpha_1 L_1 L_2+\frac{\alpha_2 L_2^2}{2}\right)\sin(\delta_1)$$
$$+s_F\cdot(-\sin\Psi_F)(\alpha_1 L_1+\alpha_2 L_2), \tag{A.16}$$
$$\frac{\partial Y_F}{\partial L_2}=\cos\left(\alpha_1 L_1 L_2+\frac{\alpha_2 L_2^2}{2}\right)\sin(\delta_1)$$
$$+\sin\left(\alpha_1 L_1 L_2+\frac{\alpha_2 L_2^2}{2}\right)\cos(\delta_1)$$
$$+s_F\cdot(\cos\Psi_F)(\alpha_1 L_1+\alpha_2 L_2), \tag{A.17}$$
$$\frac{\partial\Psi_F}{\partial L_2}=\alpha_1 L_1+\alpha_2 L_2, \tag{A.18}$$
$$\frac{\kappa_F}{\partial L_2}=\alpha_2. \tag{A.19}$$

A.5. Components with respect to s_0

$$\partial X_F/\partial s_0=1, \tag{A.20}$$
$$\partial Y_F/\partial s_0=0, \tag{A.21}$$
$$\partial\Psi_F/\partial s_0=0, \tag{A.22}$$
$$\partial\kappa_F/\partial s_0=0. \tag{A.23}$$

A.6. Components with respect to s_F

$$\partial X_F/\partial s_F=\cos(\Psi_F), \tag{A.24}$$
$$\partial Y_F/\partial s_F=\sin(\Psi_F), \tag{A.25}$$
$$\partial\Psi_F/\partial s_F=0, \tag{A.26}$$
$$\partial\kappa_F/\partial s_F=0. \tag{A.27}$$

APPENDIX B: POSE OPERATORS

These operators are defined to simplify working with rigid bodies in two dimensions. A rigid body in a plane has $2+1$ parameters, two for translation in the plane and one for heading angle. The position in metres and the heading in radians can be assembled into a three vector, called the pose.

$$\boldsymbol{p_1}=[x_1,y_1,\psi_1]^T, \tag{B.1}$$
$$\boldsymbol{p_2}=[x_2,y_2,\psi_2]^T. \tag{B.2}$$

Working with higher dimensional spaces it is common to use homogeneous coordinates to simplify operations on rigid bodies. The two operators defined here are convenient for working in 2D without the complexity of homogeneous notation. They are inverse operators in the sense that $\boldsymbol{p_3}=\boldsymbol{p_1}\oplus\boldsymbol{p_2}\rightarrow\boldsymbol{p_3}\ominus\boldsymbol{p_2}=\boldsymbol{p_1}$. The decompose operator $\ominus$ is order dependent like ordinary subtraction. They are defined below using the algebra of homogeneous coordinates given in [52].

B.2. Compose $\oplus$

This operator takes the second pose and rotates it into the frame of the first before vector addition of all three components.

$$\boldsymbol{p_3}=\boldsymbol{p_1}\oplus\boldsymbol{p_2}=\begin{bmatrix}x_1+x_2\cos(\psi_2)-y_2\sin(\psi_2)\\ y_1+x_2\sin(\psi_2)+y_2\cos(\psi_2)\\ \psi_1+\psi_2\end{bmatrix}. \tag{B.3}$$

If each pose is expressed in homogeneous 3x3 form,

$$\boldsymbol{T_1}=\begin{bmatrix}\cos(\psi_1) & -\sin(\psi_1) & x_1\\ \sin(\psi_1) & \cos(\psi_1) & y_1\\ 0 & 0 & 1\end{bmatrix}. \tag{B.4}$$

The same operation is a matrix multiplication

$$\boldsymbol{T_3}=\boldsymbol{T_1}\boldsymbol{T_2}. \tag{B.5}$$

B.3. Decompose $\ominus$

This operator finds the pose of $\boldsymbol{p_3}$ expressed relative to pose $\boldsymbol{p_1}$

$$\boldsymbol{p_1}=\boldsymbol{p_3}\ominus\boldsymbol{p_2}. \tag{B.6}$$

If each pose is expressed in homogeneous 3x3 form, the same operation is a matrix multiplication by the the inverse

$$\boldsymbol{T_1}=\boldsymbol{T_3}\boldsymbol{T_2}^{-1}. \tag{B.7}$$

REFERENCES

[1] V. Digani, F. Caramaschi, L. Sabattini, C. Secchi and C. Fantuzzi "Obstacle avoidance for industrial AGVs," *Proc. of IEEE 10th International Conference on Intelligent Computer Communication and Processing, ICCP 2014*, pp-227-232, 2014.

[2] B. Paden, M. Cap, S. Z. Yong, D. Yershov, and E. Frazzoli, "A survey of motion planning and control techniques for self-driving urban vehicles," http://arxiv.org/abs/1604.07446, pp. 1-27, 2016.

[3] I. F. Vis, "Survey of research in the design and control of automated guided vehicle systems," *European Journal of Operational Research*, vol. 179, no. 3, pp. 677-709, 2006.

[4] N. Boysen, R. de Koster, and F. Weidinger, " Warehousing in the e-commerce era - A Survey," *European Journal of Operational Research*, vol. 277, no. 2, pp. 396-411, 2019.

[5] S. M. LaValle and D. Leidner, "Motion Planning," in *Planning Algorithms*, ch. 3-8, pp. 81-412, Cambridge University Press, 2006

[6] C. Katrakazas, M. Quddus, W.-H. Chen, and L. Deka, "Real-time motion planning methods for autonomous on-road driving: State-of-the-art and future research directions," *Transportation Research Part C: Emerging Technologies*, vol. 60. pp. 416-442, 2015.

[7] J. Moreau, P. Melchior, S. Victor, L, Cassany, M. Moze, F. Aioun, and F. Guillemard, "Reactive path planning in intersection for autonomous vehicle," *IFAC-PapersOnLine*, vol. 52, no. 5, pp. 109-114, 2019.

[8] A. Chebly, R. Talj, A. Charara, A. Chebly, R. Talj, A. Charara, M. Planning, C. Alia, T. Reine, and C. Ali, "Maneuver planning for autonomous vehicles with clothoid tentacles for local trajectory planning," *Proc. of IEEE 20th International Conference on Intelligent Transportation Systems (ITSC)*, 2017.

[9] E. Bertolazzi and M. Frego, "Interpolating clothoid splines with curvature continuity," *Mathematical Methods in the Applied Sciences*, vol 41, no. 4, pp. 1723-1737, 2018.

[10] E. Lambert, R. Romano, and D. Watling, "Optimal Path Planning with Clothoid Curves for Passenger Comfort," *Proceedings of the 5th International Conference on Vehicle Technology and Intelligent Transport Systems* (O. Gusikhin and M. Helfert, eds.), (Heraklion, Crete), pp. 609-615, SCITEPRESS, 2019.

[11] M. Frego, E. Bertolazzi, F. Biral, D. Fontanelli, and L. Palopoli, "Semi-analytical minimum time solutions for a vehicle following clothoid-based trajectory subject to velocity constraints," *Proc. of European Control Conference, ECC 2016*, pp. 2221-2227, 2017.

[12] V. Girbés, L. Armesto, and J. Tornero, "Path following hybrid control for vehicle stability applied to industrial forklifts," *Robotics and Autonomous Systems*, vol. 62, no. 6, pp. 910-922, 2014.

[13] M. Pivtoraiko, R. A. Knepper, and A. Kelly, "Differentially constrained mobile robot motion planning in state lattices," *Journal of Field Robotics*, vol. 26, no. 3, pp. 308-333, 2009.

[14] B. Siciliano and O. Khatid, eds., *Springer Handbook of Robotics*, Springer Handbooks, 2nd ed., Springer, Berlin, 2016.

[15] Y. Kuwata, J. Teo, G. Fiore, S. Karaman, E. Frazzoli, and J. P. How, "Real-time motion planning with applications to autonomous urban driving," *IEEE Transactions on Control Systems Technology*, vol. 17, no. 5, pp. 1105-1118, 2009.

[16] S. M. laValle and J. J. Kuffner, "Rapidly-exploring random trees: Progress and prospects," *Proc. of 4th Workshop on Algorithmic and Computational Robotics: New Directions*, pp. 293-308, 2000.

[17] S. Yoon, D. Lee, J. Jung, and D. H. Shim, "Spline-based RRT* using piecewise continuous collision-checking algorithm for car-like vehicles," *Journal of Intelligent and Robotic Systems: Theory and Applications*, vol. 90, mo. 3-4, pp. 537-549, 2018.

[18] R. Deits and R. Tedrake, "Efficient mixed-integer planning for UAVs in cluttered environments," *Proc. of IEEE International Conference on Robotics and Automation (ICRA)*, pp. 42-49, 2015.

[19] C. Rosmann, F. Hoffmann, and T. Bertram, "Kinodynamic trajectory optimization and control for car-like robots," *Proc. of IEEE International Conference on Intelligent Robots and Systems*, vol. 2017-Septe, pp. 5681-5686, 2017.

[20] J. C. Kim, D. S. Pae, and M. T. Lim, "Obstacle avoidance path planning based on output constrained model predictive control," *International Journal of Control, Automation and Systems*, vol. 17, no. 11, pp. 2850-2861, 2019.

[21] M. Yue, X. Wu, L. Guo, and J. Gao, "Quintic polynomial-based obstacle avoidance trajectory planning and tracking control framework for tractor-trailer system," *International Journal of Control, Automation and Systems*, vol. 17, no. 10, pp. 2634-2646, 2019.

[22] R. Levien, "The Elastica: A Mathematical History," Tech. Rep., University of California, Bekeley, 2008.

[23] S. Gim, L. Adouane, S. Lee, and J. P. Dérutin, "Clothoids composition method for smooth path generation of car-like vehicle navigation,", *Journal of Intelligent and Robotic Systems: Theory and Applications*, vol. 88, no. 1, pp. 129-146, 2017.

[24] J. Henrie and D. Wilde, "Planning Continuous Curvature Paths Using Constructive Polylines," *Journal of Aerospace Computing, Information, and Communication*, vol. 4, no. 12, pp. 1143-1157, 2007.

[25] *Path Planning for Autonomous Vehicles Using Clothoid Based Smoothing of A* Generated Paths and Optimal Control*, PhD Thesis, KTH Royal Institute of Technology, 2017.

[26] T. Fraichard and A. Scheuer, "From reeds and shepp's to continuous-curvature paths," *IEEE Transactions on Robotics*, vol. 20, no. 6, pp. 1025-1035, 2004.

[27] F. Lamiraux and J. P. Laumond, "Smooth motion planning for car-like vehicles," *IEEE Transactions on Robotics and Automation*, vol. 17, no. 4, pp. 498-502, 2001.

[28] M. Fliess, J. Levine, P. Martin, and P. Rouchon, "Flatness and defect of non-linear systems: Introductory theory and examples," *International Journal of Control*, vol. 61, no. 6, pp. 1327-1361, 1995.

[29] J. A. Reeds and L. A. Shepp, "Optimal paths for a car that goes forwards and backwards," *Pacific Journal of Mathematics*, vol. 145, no. 2, pp. 367-393, 1990.

[30] S. Gim, *Flexible and Smooth Trajectory Generation Based on Parametric Clothoids for Nonholonomic Car-like Vehicles*, PhD Thesis, Université Clermon Auvergne, 2017.

[31] R. Gobithaasan, Y. Wei, K. Miura, and M. Shanmugavel, "Optimal path smoothing with log-aesthetic curves based on shortest distance, minimum bending energy and curvature variation energy," *Proceedings of CAD'19,* (Singapore), pp. 397-402, 2019.

[32] J. E. M. Solanes, L. Armesto, J. Tornero, P. Muñoz-Benavent, and V. Girbés, "Mobile robot obstacle avoidance based on quasi-holonomic smooth paths," *Proc. of Joint of the 13th Annual Conference on Towards Autonomous Robotic Systems, TAROS 2012 and the 15th Annual FIRA RoboWorld Congress*, vol. 7429, pp. 152-163, 2012.

[33] J. A. Silvan and V. Grassi, "Clothoid-based global path planning for autonomous vehicles in urban scenarios," *Proceedings - IEEE International Conference on Robotics and Automation*, pp. 4312-4318, 2018.

[34] P. F. Lima, M. Trincavelli, J. Martensson, and B. Wahlberg, "Clothoid-based model predictive control for autonomous driving," *Proc. of European Control Conference, ECC 2015,* pp. 2983-2990, 2015.

[35] M. G. Plessen, P. F. Lima, J. Martensson, A. Bemporad, and B. Wahlberg, "Trajectory planning under vehicle dimension constraints using sequential linear programming," *Proc. of IEEE Conference on Intelligent Transportation Systems, Proceedings, ITSC*, vol. 2018-March, pp. 1-6, 2018.

[36] D. Shin, S. Singh, and W. Wittaker, "Path generation for a robot vehicle using composite clothoid segments," *IFAC Proceedings Volumes*, vol. 25, no. 6, pp. 443-448, 1992.

[37] D. J. Walton and D. S. Meek, "A controlled clothoid spline," *Computers and Graphics*, vol. 29, no. 3, pp. 353-363, 2005.

[38] S. Thrun and A. Buecken, "Integrating grid-based and topological maps for mobile robot navigation," *Proceedings of the National Conference on Artificial Intelligence*, vol. 2, no. August, pp. 944-950, 1996.

[39] S. Thrun, W. Burgard, and D. Fox, *Probabilistic Robotics*, MIT Press, Cambridge, Mass., 2005.

[40] S. M. LaValle and D. Leidner, "Chapter 6: Combinatorial Motion Planning," in *Planning Alogrithms,* ch. 6, pp. 249-310, Cambridge University Press, 2006.

[41] R. Deits and R. Tedrake, "Computing large convex regions of obstacle-free space through semidefinite programming," *Springer Tracts in Advanced Robotics*, vol. 107, pp. 109-124, 2015.

[42] T. Gu, *Improved Trajectory Planning for On-Road Self-Driving Vehicles Via Combined Graph Search, Optimization & Topology Analysis*, Doctor of Philisophy, Carnegie Mellon University, 2017.

[43] Hyster, "E30-40HSD Technical Guide," www.hyster.com, August 2020.

[44] R. Baird, *An Autonomous Forklift Research Platform for Warehouse Operations*, Master's thesis, Massachusetts Institude of Technology, 2018.

[45] G. Raballan, "How do differing standards increase trade costs? THE CASE OF PALLETS Ga"el," *World Bank Policy Research Working Paper 3519,* pp. 1-20, 2005.

[46] R. Bostelman, R. Nocross, J. Falco, and J. Marvel, "Development of standard test methods for unmanned and manned industrial vehicles used near humans," *Multisensor, Multisource Information Fusion: Architectures, Algorithms, and Applications*, vol. 8756, p. 87560P, 2013.

[47] M. P. Kelly, "Transcription Methods for Trajectory Optimization: a beginners tutorial," Tech. Rep., 2017. http://arxiv.org/abs/1707.00284

[48] "MATLAB Help Center - Choosing the Algorithm.," https://uk.mathworks.com/help/optim/ug/choosing-the-algorithm.html, August 2020.

[49] L. F. Shampine, "Vectorized adaptive quadrature in MATLAB," *Journal of Computational and Applied Mathematics*, vol. 211, no. 2, pp. 131-140, 2008.

[50] M. A. Cooper, J. F. Raquet, and R. Patton, "Range information characterization of the Hokuyo UST-20LX LIDAR sensor," *Phtonics,* vol. 5, no. 2, 2018.

[51] M. Grant and S. Boyd, "CVX: Matlab Software for Disciplined Convex Programming, version 2.1.," http://cvxr.com/cvx

[52] K. J. Waldron and J. Schmiedeler, "Kinematics," in *Springer Handbook of Robotics*, ch. 2, pp. 11–36, Springer, 2016.

Edward Derek Lambert received his MEng in Engineering Science from the University of Oxford, graduating with a first class degree in 2013. After some time in industry, he began an EPSRC Doctoral Training Partnership Studentship in 2018, working towards a PhD degree at the Institute for Transport Studies at the University of Leeds. His research interests include optimisation-based path planning and multiple vehicle motion coordination.

Richard Romano has over twenty five years of experience developing and testing AVs and ADAS concepts and systems which began with the Automated Highway Systems (AHS) project while he directed the Iowa Driving Simulator in the early 1990's. He recieved his BASc and MASc in Engineering Science and Aerospace Engineering respectively from the University of Toronto, Canada and a PhD in Motion Drive Algorithms for Large Excursion Motion Bases, Industrial Engineering from the University of Iowa, USA. In addition to a distinguished career in industry he has supervised numerous research projects and authored many journal papers. In 2015 he was appointed a Professor of Driving Simulation at the Institute for Transport Studies, University of Leeds, UK. His research interests include the development, validation and application of transport simulation to support the human-centred design of vehicles and infrastructure.

David Watling's primary research focus is the development of mathematical models and methods for analysing transport systems, especially those that represent the interactions between travellers' decision-making and the physical infrastructure. He has particularly developed methods for modelling, simulating or optimizing transport networks with random, dynamic or unreliable elements. With a B.Sc. degree in mathematics from the University of Leeds and a Ph.D. from the Department of Probability and Statistics at the University of Sheffield, he has held the post of Centenary Chair of Transport Analysis at the University of Leeds since its instigation in 2004, where he is the co-leader of the Spatial Modelling and Dynamics group in the Institute for Transport Studies.

A.1 Bits of LaTeXadvice

1. Do look at the output log and try to understand any errors - they are sometimes important!

2. In the final pdf, do a search for ? - it is what LaTeXwill give when a reference is missing. Having missing references in your submitted thesis is, at best, embarrassing and potentially a failing matter.

3. A good quality bib file is important - make sure that entries are consistent in whether journals are abreviated, capitalised and how Author names are presented. A good way to do this is to use Mendeley to import your bib file and then use its doi lookup feature which will re-write your bibliogrpahy entries in a standardised form. You then export the bibliography back oit as a bib file.

4. Be particularly careful about older papers where the doi may not be easy to track down. Also watch out for JETP Letters that you are being consistent in citing the English language version (or the Russian, but don't mix and match!)

5. Although LaTeXguides may show you how to assemble a multi-part fgure from within LaTeX, it can be hard to make sub-plots appear exactly the same size. We recommend using something like Inkscape to assemble the parts of a figure and lay them out nicely. Be careful if saving to pdf files thatr the fonts are preserved - otherwise you can lose greek symbols.

6. If preparing figures in Origin, set the plot size to be exactly the right size or exactly double size and then scale fonts and symbols accordingly. Use Origin's ability to copy formatting between graphs to make everything nicely consistent (e.g. frame sizes, thicknesses, coolour schemes, point sizes and shapes).

7. In general resist the temptation to put [H] when placing figures and tables - in most cases it is better to let LaTeXwork out where to put things. It can get tricky if you have a lot of figures one after another (perhaps a single multi-part figure is what you need?) - the placement option [p] can alos help to move floats to a separate page of figures. See also the *afterpage* package.

Appendix B

Intersection Control

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.

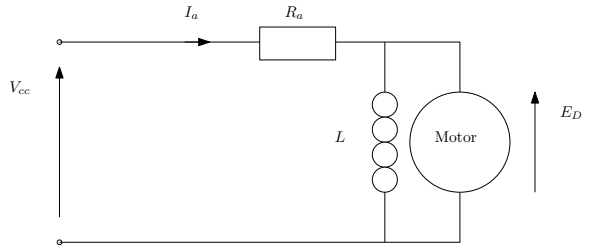


Fig. 1 Steady state equivalent circuit for a DC motor.

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 = 1m 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 **??**

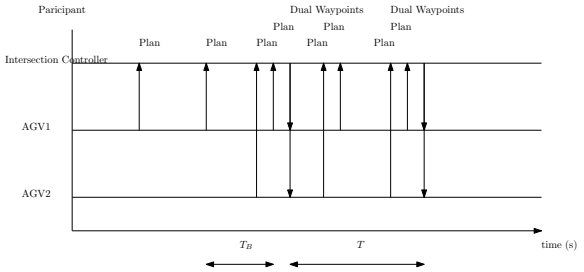


Fig. 2 Sequence diagram for two AGVs communicating with the Intersection Controller which sends a DualWaypoint message to every known AGV every T seconds, considering the latest ApproachPlans it has recieved to date.

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 $\boldsymbol{d}_i = [d_{AB}, d_{BC}]$.

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

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

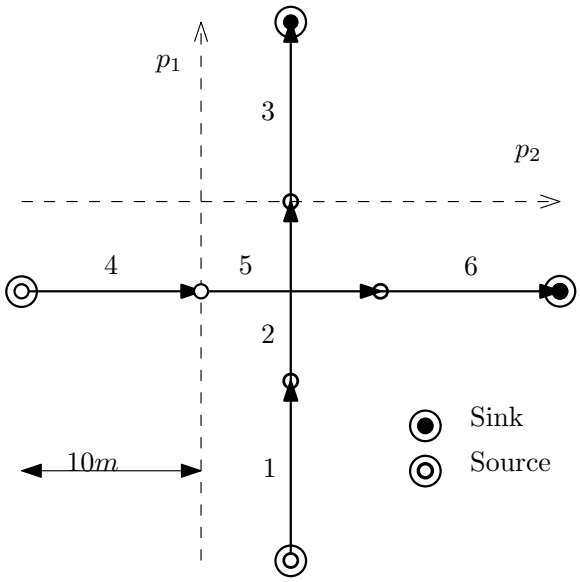


Fig. 3 Intersection layout with two conflicting routes.

	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.

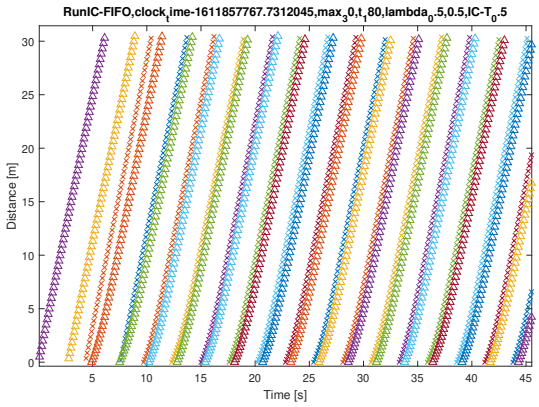


Fig. 4 Position-Time trace for HLHT Scenario under FIFO controller

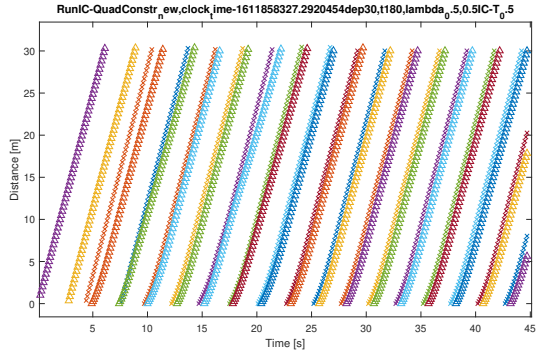


Fig. 5 Position-Time trace for HLHT Scenario under Quadratic Constraints controller

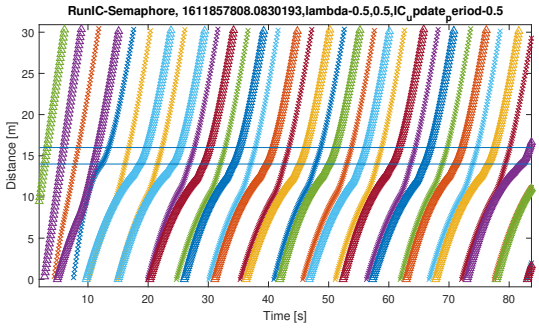


Fig. 6 Position-Time trace for HLHT Scenario under Semaphore controller

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

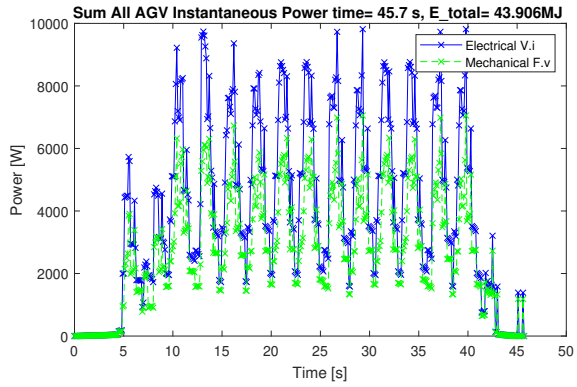


Fig. 7 Power Dissipation-Time trace for HLHT Scenario under FIFO controller

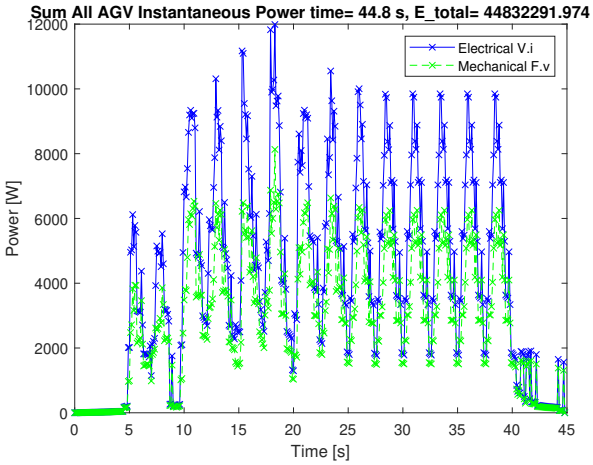


Fig. 8 Power Dissipation-Time trace for HLHT Scenario under Quadratic Constraints controller

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

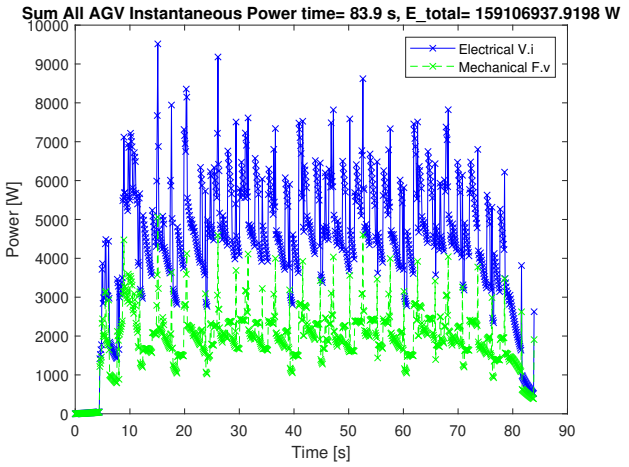


Fig. 9 Power Dissipation-Time trace for HLHT Scenario under Semaphore controller

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

Bibliography

[1] Bruno Siciliano and Oussama Khatib, editors. *Springer Handbook of Robotics*. Springer Handbooks. Springer, Berlin, 2 edition, 2016. ISBN 978-3-319-32550-7. doi: 10.1007/978-3-540-30301-5.

[2] Wilko Schwarting, Javier Alonso-Mora, and Daniela Rus. Planning and Decision-Making for Autonomous Vehicles. *Annual Review of Control, Robotics, and Autonomous Systems*, 1(1):annurev—-control—-060117—-105157, 2018. ISSN 2573-5144. doi: 10.1146/annurev-control-060117-105157. URL http://www.annualreviews.org/doi/10.1146/annurev-control-060117-105157.

[3] Chris Urmson. Autonomous Driving in Urban Environments: Boss and the Urban Challenge. *Journal of Field Robotics*, 25(8):425–466, 2008. ISSN 14746670. doi: 10.1002/rob.

[4] Lu Chi and Yadong Mu. Deep Steering: Learning End-to-End Driving Model from Spatial and Temporal Visual Cues. *CoRR*, abs/1708.0, 2017. URL http://arxiv.org/abs/1708.03798.

[5] Changxi You, Jianbo Lu, Dimitar Filev, and Panagiotis Tsiotras. Advanced planning for autonomous vehicles using reinforcement learning and deep inverse reinforcement learning. *Robotics and Autonomous Systems*, 114:1–18, 2019. ISSN 09218890. doi: 10.1016/j.robot.2019.01.003. URL https://doi.org/10.1016/j.robot.2019.01.003.

[6] Robert Walenta, Twan Schellekens, Alexander Ferrein, and Stefan Schiffer. A decentralised system approach for controlling AGVs with ROS. *2017 IEEE AFRICON: Science, Technology and Innovation for Africa, AFRICON 2017*, pages 1436–1441, 2017. doi: 10.1109/AFRCON.2017.8095693.

[7] Anis Koubaa. *Robot Operating System (ROS): The Complete Reference (Volume 1)*. Springer Publishing Company, Incorporated, 1st edition, 2016. ISBN 3319260529, 9783319260525.

[8] Keonyup Chu, Minchae Lee, and Myoungho Sunwoo. Local path planning for off-road autonomous driving with avoidance of static obstacles. *IEEE Transactions on Intelligent Transportation Systems*, 13(4):1599–1616, 2012. ISSN 15249050. doi: 10.1109/TITS.2012.2198214.

[9] Valerio Digani, Fabrizio Caramaschi, Lorenzo Sabattini, Cristian Secchi, and Cesare Fantuzzi. Obstacle avoidance for industrial AGVs. *Proceedings - 2014 IEEE 10th International Conference on Intelligent Computer Communication and Processing, ICCP 2014*, pages 227–232, 2014. doi: 10.1109/ICCP.2014.6937001.

[10] Suhyeon Gim, Lounis Adouane, Sukhan Lee, and Jean Pierre Dérutin. Clothoids Composition Method for Smooth Path Generation of Car-Like Vehicle Navigation. *Journal of Intelligent and Robotic Systems: Theory and Applications*, 88(1):129–146, 2017. ISSN 15730409. doi: 10.1007/s10846-017-0531-8.

[11] Joshua Henrie and Doran Wilde. Planning Continuous Curvature Paths Using Constructive Polylines. *Journal of Aerospace Computing, Information, and Communication*, 4(12):1143–1157, 2007. doi: 10.2514/1.32776.

[12] Doran K Wilde. Computing clothoid segments for trajectory generation. *2009 IEEE/RSJ International Conference on Intelligent Robots and Systems, IROS 2009*, pages 2440–2445, 2009. doi: 10.1109/IROS.2009.5354700.

[13] Steven M. LaValle and James J. Kuffner. Rapidly-exploring random trees: Progress and prospects. *4th Workshop on Algorithmic and Computational Robotics: New Directions*, pages 293–308, 2000. ISSN 16130073. doi: 10.1017/CBO9781107415324.004. URL http://citeseerx.ist.psu.edu/viewdoc/summary?doi=10.1.1.38.1387.

[14] Marcus Lundberg. Path planning for autonomous vehicles using clothoid based smoothing of A * generated paths and optimal control. 2017.

[15] D. J. Walton and D. S. Meek. A controlled clothoid spline. *Computers and*

Graphics (Pergamon), 29(3):353–363, 2005. ISSN 00978493. doi: 10.1016/j.cag.2005.03.008.

[16] Mikhail Pivtoraiko, Ross A Knepper, and Alonzo Kelly. Differentially constrained mobile robot motion planning in state lattices. *Journal of Field Robotics*, 26(3): 308–333, 2009.

[17] Elena Cardarelli, Valerio Digani, Lorenzo Sabattini, Cristian Secchi, and Cesare Fantuzzi. Cooperative cloud robotics architecture for the coordination of multi-AGV systems in industrial warehouses. *Mechatronics*, 45:1–13, 2017. ISSN 09574158. doi: 10.1016/j.mechatronics.2017.04.005.

[18] Valerio Digani, Lorenzo Sabattini, Cristian Secchi, and Cesare Fantuzzi. Ensemble Coordination Approach in Multi-AGV Systems Applied to Industrial Warehouses. *IEEE Transactions on Automation Science and Engineering*, 12(3):922–934, 2015. ISSN 15455955. doi: 10.1109/TASE.2015.2446614.

[19] Iris F.A. Vis. Survey of research in the design and control of automated guided vehicle systems. *European Journal of Operational Research*, 170(3):677–709, 2006. ISSN 03772217. doi: 10.1016/j.ejor.2004.09.020.

[20] Mariagrazia Dotoli, Alexander Fay, Marek Miśkowicz, and Carla Seatzu. An overview of current technologies and emerging trends in factory automation. *International Journal of Production Research*, 57(15-16):5047–5067, 2019. ISSN 1366588X. doi: 10.1080/00207543.2018.1510558. URL https://doi.org/10.1080/00207543.2018.1510558.

[21] Valerio Digani, M. Ani Hsieh, Lorenzo Sabattini, and Cristian Secchi. Coordination of multiple AGVs: a quadratic optimization method. *Autonomous Robots*, 43(3): 539–555, 2019. ISSN 15737527. doi: 10.1007/s10514-018-9730-9. URL https://doi.org/10.1007/s10514-018-9730-9.

[22] Kumiko Tadano and Yoshiharu Maeno. Scalable control system for dense transfer vehicles in flexible manufacturing. *2019 IEEE Conference on Control Technology and Applications (CCTA)*, (3):325–331, 2019.

[23] Ivica Draganjac, Tamara Petrović, Damjan Miklić, Zdenko Kovačić, and Juraj Oršulić. Highly-scalable traffic management of autonomous industrial trans-

portation systems. *Robotics and Computer-Integrated Manufacturing*, 63(July 2018):101915, 2020. ISSN 07365845. doi: 10.1016/j.rcim.2019.101915. URL https://doi.org/10.1016/j.rcim.2019.101915.

[24] Toshiyuki Miyamoto and Kensuke Inoue. Local and random searches for dispatch and conflict-free routing problem of capacitated AGV systems. *Computers and Industrial Engineering*, 91:1–9, 2016. ISSN 03608352. doi: 10.1016/j.cie.2015.10.017. URL http://dx.doi.org/10.1016/j.cie.2015.10.017.

[25] Bai Li, Hong Liu, Duo Xiao, Guizhen Yu, and Youmin Zhang. Centralized and optimal motion planning for large-scale AGV systems: A generic approach. *Advances in Engineering Software*, 106:33–46, 2017. ISSN 18735339. doi: 10.1016/j.advengsoft.2017.01.002. URL http://dx.doi.org/10.1016/j.advengsoft.2017.01.002.

[26] Jur P. Van Den Berg and Mark H. Overmars. Prioritized motion planning for multiple robots. *2005 IEEE/RSJ International Conference on Intelligent Robots and Systems, IROS*, pages 2217–2222, 2005. doi: 10.1109/IROS.2005.1545306.

[27] Arthur Richards and Jonathan P. How. Aircraft Trajectory Planning With Collision Avoidance Using Mixed Integer Linear Programming. In *Proceedings of the American Control Conference*, pages 1936–1941, Anchorage, AK, USA, 2002. ISBN 0000000000.

[28] M. Dotoli and M. P. Fanti. Coloured timed Petri net model for real-time control of automated guided vehicle systems. *International Journal of Production Research*, 42(9):1787–1814, 2004. ISSN 00207543. doi: 10.1080/00207540410001661364.

[29] S. P. Singh and M. K. Tiwari. Intelligent agent framework to determine the optimal conflict-free path for an automated guided vehicles system. *International Journal of Production Research*, 40(16):4195–4223, 2002. ISSN 00207543. doi: 10.1080/00207540210155783.

[30] G Sanchez and J Latombe. Using a PRM Planner to Compare centralized and decoupled planning for multi-robot systems. In *IEEE International Conference on Robotics and Automation*, pages 2112–2119, 2002.

[31] Mike Peasgood, Christopher M. Clark, and John McPhee. A complete and scalable strategy for coordinating multiple robots within roadmaps. *IEEE Transactions on Robotics*, 24(2):283–292, 2008. ISSN 15523098. doi: 10.1109/TRO.2008.918056.

[32] Jingjin Yu and Steven M. Lavalle. Planning optimal paths for multiple robots on graphs. *Proceedings - IEEE International Conference on Robotics and Automation*, pages 3612–3617, 2013. ISSN 10504729. doi: 10.1109/ICRA.2013.6631084.

[33] Jingjin Yu and Daniela Rus. An Effective Algorithmic Framework for Near Optimal Multi-robot Path Planning. pages 495–511, 2018. doi: 10.1007/978-3-319-51532-8_30.

[34] Lubomír Bakule. Decentralized control: An overview. *Annual Reviews in Control*, 32(1):87–98, 2008. ISSN 13675788. doi: 10.1016/j.arcontrol.2008.03.004.

[35] Ivica Draganjac, Damjan Miklic, Zdenko Kovacic, Goran Vasiljevic, and Stjepan Bogdan. Decentralized Control of Multi-AGV Systems in Autonomous Warehousing Applications. *IEEE Transactions on Automation Science and Engineering*, 13 (4):1433–1447, 2016. ISSN 15455955. doi: 10.1109/TASE.2016.2603781.

[36] Lorenzo Sabattini, Mika Aikio, Patric Beinschob, Markus Boehning, Elena Cardarelli, Valerio Digani, Annette Krengel, Massimiliano Magnani, Szilard Mandici, Fabio Oleari, Christoph Reinke, Davide Ronzoni, Christian Stimming, Robert Varga, Andrei Vatavu, Sergi Castells Lopez, Cesare Fantuzzi, Aki Mayra, Sergiu Nedevschi, Cristian Secchi, and Kay Fuerstenberg. The PAN-robots project: Advanced automated guided vehicle systems for industrial logistics. *IEEE Robotics and Automation Magazine*, 25(1):55–64, 2018. ISSN 10709932. doi: 10.1109/MRA.2017.2700325.

[37] Lorenzo Sabattini, Valerio Digani, Matteo Lucchi, Cristian Secchi, and Cesare Fantuzzi. Mission Assignment for Multi-Vehicle Systems in Industrial Environments. *IFAC-PapersOnLine*, 48(19):268–273, 2015. ISSN 24058963. doi: 10.1016/j.ifacol.2015.12.044. URL http://dx.doi.org/10.1016/j.ifacol.2015.12.044.

[38] Charlie Street, Bruno Lacerda, Manuel Mühlig, and Nick Hawes. Multi-Robot Planning Under Uncertainty with Congestion-Aware Models. In N. B. An, A. Yorke-Smith, El Fallah Seghrouchni, and G. Sukthankar, editors, *Proc. of the*

19th International Conference on Autonomous Agents and Multiagent Systems, page 9, Auckland, New Zealand, 2020.

[39] Jackeline Rios-Torres and Andreas A. Malikopoulos. A Survey on the Coordination of Connected and Automated Vehicles at Intersections and Merging at Highway On-Ramps. *IEEE Transactions on Intelligent Transportation Systems*, 18(5):1066–1077, 2017. ISSN 15249050. doi: 10.1109/TITS.2016.2600504.

[40] Kurt Dresner. A Multiagent Approach to Autonomous Intersection Management. *Journal of Artificial Intelligence Research*, 31:591–656, 2008.

[41] Andreas A. Malikopoulos, Christos G. Cassandras, and Yue J. Zhang. A decentralized energy-optimal control framework for connected automated vehicles at signal-free intersections. *Automatica*, 93:244–256, 2018. ISSN 00051098. doi: 10.1016/j.automatica.2018.03.056. URL https://doi.org/10.1016/j.automatica.2018.03.056.

[42] Yu Zhang, Huiyan Chen, Steven L. Waslander, Jianwei Gong, Guangming Xiong, Tian Yang, and Kai Liu. Hybrid Trajectory Planning for Autonomous Driving in Highly Constrained Environments. *IEEE Access*, 6:32800–32819, 2018. ISSN 21693536. doi: 10.1109/ACCESS.2018.2845448.

[43] Michael Alexander Djojo and Kanisius Karyono. Computational load analysis of Dijkstra, A*, and Floyd-Warshall algorithms in mesh network. *Proceedings of 2013 International Conference on Robotics, Biomimetics, Intelligent Computational Systems, ROBIONETICS 2013*, pages 104–108, 2013. doi: 10.1109/ROBIONETICS.2013.6743587.

[44] Szymon Racewicz, Paweł Kazimierczuk, Bronisław Kolator, and Andrzej Olszewski. Use of 3 kW BLDC motor for light two-wheeled electric vehicle construction. *IOP Conference Series: Materials Science and Engineering*, 421(4), 2018. ISSN 1757899X. doi: 10.1088/1757-899X/421/4/042067.

[45] Edward Hughes. Direct-current Motors. In JOHN HILEY, KEITH BROWN, and IAN McKENZIE SMITH, editors, *Electrical and Electronic Technology*, chapter 42, pages 870–883. Pearson, 10 edition, 2008. ISBN 9780132060110.

[46] Bulent Sarlioglu. Understanding Electric Motors and Loss Mechanisms. 2016. URL https://www.irc.wisc.edu/export.php?ID=421.

[47] Drag Coefficient, 2004. URL https://www.engineeringtoolbox.com/drag-coefficient-d{_}627.html.

[48] VL Knoop. Headway models, 2020. URL https://ocw.tudelft.nl/courses/traffic-flow-theory-simulation/?view=info.

[49] Yosef Sheffi. *Urban_Transportation_Networks*, volume id. 1985. ISBN 0139397299.

[50] Michael W. Levin and David Rey. Conflict-point formulation of intersection control for autonomous vehicles. *Transportation Research Part C: Emerging Technologies*, 85(January):528–547, 2017. ISSN 0968090X. doi: 10.1016/j.trc.2017.09.025. URL http://dx.doi.org/10.1016/j.trc.2017.09.025.

[51] Yuchu He, Zhijuan Jia, Yage Cheng, Zhe Li, Lipeng Wang, and Junjun Fu. Modeling and Simulation of Heterogeneous Traffic Flow in the Vicinity of Intersections Considering Communication Delay. *Chinese Control Conference, CCC*, 2020-July: 5665–5670, 2020. ISSN 21612927. doi: 10.23919/CCC50068.2020.9189519.

[52] Malachy Carey, Hillel Bar-Gera, David Watling, and Chandra Balijepalli. Implementing first-in-first-out in the cell transmission model for networks. *Transportation Research Part B: Methodological*, 65:105–118, 2014. ISSN 01912615. doi: 10.1016/j.trb.2014.04.001. URL http://dx.doi.org/10.1016/j.trb.2014.04.001.

[53] Stephen Boyd and Lieven Vandenberghe. *Chapter 3: Convex Functions*. Cambridge University Press, 7 edition, 2004. ISBN 978-0-521-83378-3. URL www.cambridge.org/9780521833783.

[54] Walter Murray and Kien Ming Ng. An algorithm for nonlinear optimization problems with binary variables. *Computational Optimization and Applications*, 47(2): 257–288, 2010. ISSN 09266003. doi: 10.1007/s10589-008-9218-1.

[55] Mark B Duinkerken, Joseph J M Evers, and Jaap A Ottjes. TRACES : TRAFFIC CONTROL ENGINEERING SYSTEM A case-study on container terminal automation General description. In *Proc. of the Summer Computer Simulation Conference*, number July, Chicago, 1999. ISBN 1565551737.

[56] C. K.M. Lee, K. L. Keung, K. K.H. Ng, and Daniel C.P. Lai. Simulation-based Multiple Automated Guided Vehicles Considering Charging and Collision-free Requirements in Automatic Warehouse. *IEEE International Conference on Industrial Engineering and Engineering Management*, 2019-Decem(December):1376–1380, 2019. ISSN 2157362X. doi: 10.1109/IEEM.2018.8607396.

[57] Suhyeon Gim. *Flexible and Smooth Trajectory Generation based on Parametric Clothoids for Nonholonomic Car-like Vehicles*. PhD thesis, Université Clermont Auvergne, 2017. URL https://tel.archives-ouvertes.fr/tel-01751530.

[58] Suhyeon Gim, Lounis Adouane, Sukhan Lee, and Jean Pierre Dérutin. Clothoids Composition Method for Smooth Path Generation of Car-Like Vehicle Navigation. *Journal of Intelligent and Robotic Systems: Theory and Applications*, 88(1):129–146, 2017. ISSN 15730409. doi: 10.1007/s10846-017-0531-8.

[59] Miguel E Vázquez-Méndez and G Casal. The Clothoid Computation: A Simple and Efficient Numerical Algorithm. *Journal of Surveying Engineering*, 142(3): 4016005, 2016. ISSN 0733-9453. doi: 10.1061/(asce)su.1943-5428.0000177.

[60] Tan Jian-an. Comparison of capacity between roundabout design and signalised junction design. *Swiss Transport Research Conference*, page 18, 2001. URL http://scholar.google.com/scholar?hl=en{&}btnG=Search{&}q=intitle:Comparison+of+capacity+between+roundabout+design+and+signalised+junction+design{#}0.

[61] Jian John Lu, Shengdi Chen, Xing Ge, and Fuquan Pan. A programmable calculation procedure for number of traffic conflict points at highway intersections. *Journal of Advanced Transportation*, 47(8):692–703, dec 2013. ISSN 01976729. doi: 10.1002/atr.190. URL http://doi.wiley.com/10.1002/atr.190.

[62] Weiming Zhao, Ronghui Liu, and Dong Ngoduy. A bilevel programming model for autonomous intersection control and trajectory planning. *Transportmetrica A: Transport Science*, 0(0):1–25, 2019. ISSN 23249943. doi: 10.1080/23249935.2018.1563921. URL https://doi.org/23249935.2018.1563921.

[63] Liuhui Zhao, Andreas Malikopoulos, and Jackeline Rios-Torres. Optimal Control of Connected and Automated Vehicles at Roundabouts: An Investigation in a Mixed-Traffic Environment. *IFAC-PapersOnLine*, 51(9):73–78, 2018. ISSN 24058963.

doi: 10.1016/j.ifacol.2018.07.013. URL https://doi.org/10.1016/j.ifacol.2018.07.013.

[64] Ezequiel Debada, Laleh Makarem, and Denis Gillet. Autonomous coordination of heterogeneous vehicles at roundabouts. *IEEE Conference on Intelligent Transportation Systems, Proceedings, ITSC*, pages 1489–1495, 2016. doi: 10.1109/ITSC.2016.7795754.

[65] Ezequiel Gonzale Debada and Denis Gillet. Virtual Vehicle-Based Cooperative Maneuver Planning for Connected Automated Vehicles at Single-Lane Roundabouts. *IEEE Intelligent Transportation Systems Magazine*, 10(4):35–46, 2018. ISSN 19411197. doi: 10.1109/MITS.2018.2867529.

[66] Biao Xu, Shengbo Eben Li, Yougang Bian, Shen Li, Xuegang Jeff Ban, Jianqiang Wang, and Keqiang Li. Distributed conflict-free cooperation for multiple connected vehicles at unsignalized intersections. *Transportation Research Part C: Emerging Technologies*, 93(December 2017):322–334, 2018. ISSN 0968090X. doi: 10.1016/j.trc.2018.06.004.

[67] Changliu Liu, Chung Wei Lin, Shinichi Shiraishi, and Masayoshi Tomizuka. Distributed Conflict Resolution for Connected Autonomous Vehicles. *IEEE Transactions on Intelligent Vehicles*, 3(1):18–29, 2018. ISSN 23798858. doi: 10.1109/TIV.2017.2788209.

[68] Gabriel Rodrigues De Campos, Paolo Falcone, Robert Hult, Henk Wymeersch, and Jonas Sjöberg. Traffic Coordination at Road Intersections: Autonomous Decision-Making Algorithms Using Model-Based Heuristics. *IEEE Intelligent Transportation Systems Magazine*, 9(1):8–21, 2017. ISSN 19391390. doi: 10.1109/MITS.2016.2630585.

[69] Tom Schouwenaars, Jonathan How, and Eric Feron. Decentralized Cooperative Trajectory Planning of Multiple Aircraft with Hard Safety Guarantees. *AIAA Guidance, Navigation, and Control Conference and Exhibit*, (August), 2004. doi: 10.2514/6.2004-5141. URL http://arc.aiaa.org/doi/10.2514/6.2004-5141.

[70] Tamás Keviczky, Francesco Borrelli, Kingsley Fregene, Datta Godbole, and Gary J. Balas. Decentralized receding horizon control and coordination of autonomous

vehicle formations. *IEEE Transactions on Control Systems Technology*, 16(1): 19–33, 2008. ISSN 10636536. doi: 10.1109/TCST.2007.903066.

[71] G. Inalhan, D.M. Stipanovic, and C.J. Tomlin. Decentralized optimization, with application to multiple aircraft coordination. *Proceedings of the 41st IEEE Conference on Decision and Control*, 1:1147–1155, 2002. ISSN 0191-2216. doi: 10.1109/CDC.2002.1184667. URL http://ieeexplore.ieee.org/lpdocs/epic03/wrapper.htm?arnumber=1184667.

[72] Li Dai, Qun Cao, Yuanqing Xia, and Yulong Gao. Distributed MPC for formation of multi-agent systems with collision avoidance and obstacle avoidance. *Journal of the Franklin Institute*, 354(4):2068–2085, 2017. ISSN 00160032. doi: 10.1016/j.jfranklin.2016.12.021. URL http://dx.doi.org/10.1016/j.jfranklin.2016.12.021.

[73] Carlos E. Luis and Angela P. Schoellig. Trajectory generation for multiagent point-to-point transitions via distributed model predictive control. *arXiv*, 4(2):375–382, 2018. ISSN 2377-3766. doi: 10.1109/lra.2018.2890572.