Optimal Acceleration Algorithm for Connected, Semi-Autonomous Vehicles

Nicolas Vergel, Charlie Vollmer
Harvey Mudd College, Sandia National Laboratories

Abstract

For a large number of vehicles to efficiently pass through an intersection from a stop, it is imperative that vehicles are controlled in a semi-autonomous fashion and through co-operative coordination. In this paper we develop an optimal acceleration algorithm for a group of vehicles communicating in platoon-like coordination. Based on well-studied experiments of natural acceleration, our algorithm enables a real-time computation of coordinated acceleration that maintains any desired safety margin for all road conditions under a "smart city/street" environment, while optimizing road throughput efficiency. Additionally, our algorithm generalizes to be optimal under any and all moments when semi-autonomous vehicles are accelerating. Moreover, our algorithm can be implemented with no change to current road infrastructure, on vehicles of the lowest level of autonomy, and will be a building block for all current state-of-the-art smart traffic control systems.

I. Introduction

Autonomous driving technologies have clearly been undergoing rapid advancements, due in part to the reduction of sensing technology costs alongside increases in mobile computational power. No longer a simple academic exercise, autonomous vehicle technology has been rapidly entering our private and personal lives through industry initiatives of the last half decade. Google recorded more than 3 million miles of autonomous driving [1], Uber has not only implemented a pilot self-driving program [2] but also promised to replace all their human drivers with autonomous vehicles [3], and Tesla's wildly successful Autopilot has been a commercial feature in their Model S cars since 2016 [4].

Apart from increased productivity from freeing the driver of the burden to control the vehicle, the enormity of the perceived societal benefits of autonomous driving technology motivates our current research. For example, Los Angeles, CA suffers some of the worst traffic in the United States, and the cost to California citizens is an estimated annual average of $2,960 dollars and 102 hours spent in traffic [5]. Across the entire United States, the economic impact of traffic congestion is estimated as high as $150 billion [6]. Originally created for horse and carriage in the early 20th century [7], the traffic signal is undoubtedly where idle vehicles spend time and continue adding to CO2 gas emissions and the estimated 20 billion USD from premature deaths resulting from pollution, in 2010 in the United States alone [8]. In the United States in 2009, traffic crashes alone were estimated to cost 300 billion USD [9], apart from the obvious personal harm being done.

Our contribution to this nascent field -and the focus of this paper- is to offer a simple solution to help some of the largest goals of Autonomous Vehicles (AVs): alleviate road congestion through higher throughput and improving road safety by removing human error from vehicle operation. While AV software systems generally encompass three core competencies -*perception, planning,* and *control*- our work focuses on a narrow sub-field within planning and control which has not garnered much research attention: acceleration in real-time multi-vehicle motion planning. In this paper we develop an algorithm for optimal acceleration of a group of semi-autonomous vehicles, communicating and acting in a cooperative, coordinated manner. Our algorithm, however, does not have constraints found in the more general cases of vehicles having *independent goals* [10] or needing complex path planning as in [11] or [12].

While developed and illustrated on vehicles at an intersection, our algorithm easily generalizes to enable a group of vehicles to maximize throughput at any position in the roadway where vehicles find themselves accelerating. Additionally, our algorithm explicitly accounts for any desired level of safety, dynamically adjusts for changes in road conditions, and can be implemented using simple

Preliminary work, still in progress. Do not distribute.

vehicle connectivity and communication. We are able to achieve this due to the recent advances in the study of multi-vehicle cooperation, such as vehicle platooning [18], and the advances in vehicle-to-vehicle (V2V) communication technology, from the software development side in cooperative perception [13] that have been experimentally validated [14] to the hardware advances and off-the-shelf components currently for sale on the market [15].

To illustrate the effect of this simple delay from human reaction times on vehicle motion, recall that once a stoplight turns green, the first -and only- car to accelerate is the one in the front. All vehicles behind it will wait some non-zero time before they begin to accelerate, causing a dissonance between the ideal response and their actual reaction. This causes a "slinky" effect, where the further back in the queue, the longer the driver waits before accelerating. The effect is further exacerbated by our inability to safely match the acceleration of the vehicle immediately in front of us.

Our acceleration algorithm presented in this paper improves intersection efficiency by taking advantage of queued platoons of vehicles at a signalized intersection that guarantees maximum flow rates through simultaneous acceleration, while ensuring safe driving conditions. The diagram in Figure 1 illustrates how our acceleration scheme compares to the "slinky effect". Of important note, however, is that our algorithm will seamlessly work as a building block in state-of-the-art continuous flow intersections as described in [21] and [22]. Importantly, our algorithm integrates as a fundamental component in more simple V2V situations that are implementable under current semi-autonomous vehicle agency, as well as more complex Vehicle to Infrastructure (V2I) communications as a component in a centralized vehicle coordination algorithm. In addition to personal AV situations, our algorithm can be implemented for all multi-agent roadway and rail situations, such as autonomous truck convoy systems [16].

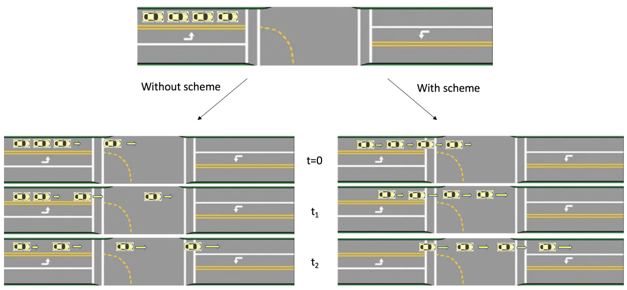

Fig. 1: On the left side, cars begin to accelerate at different times due to human delay. The right side shows a platoon of cars under our acceleration algorithm.

II. Acceleration Algorithm

In this section, we completely derive the optimal algorithm of acceleration for a group of semi-autonomous vehicles queued at an intersection in a platoon-like fashion. Our acceleration algorithm is derived from first principles of physical motion defined under discrete acceleration. We then extend the derivation to a realistic, continuous acceleration curve which has been experimentally studied to follow natural human acceleration patterns. Throughout the following derivations, we optimize each set of equations under the constraint of maximizing the number of vehicles to pass through an intersection while guaranteeing any desired level of safety for passengers.

A. Derivation of a Mean Acceleration

Without loss of generality, the following construction is for a platoon of n cars queued at an intersection. Let car_i represent the i-th car in the platoon and assume that once the light turns green, all cars have constant acceleration and each car reaches the speed limit, which we denote v_f for final velocity. Now let v_i and a_i represent the velocity and acceleration of car_i. Given that all cars are queued at a stoplight, the initial velocity is zero. Thus, the expressions for velocity and displacement simplify to the following universal equations:

$$v_i = a_i t \tag{1}$$

and

$$s_i = \frac{1}{2} a_i t^2 \tag{2}$$

where s_i is the displacement of car_i and t is time elapsed after the light turns green.

We further define τ_i to be the time it takes car_i to achieve v_f from v_0. Therefore, τ_i is the minimum value of t that satisfies the following equation:

$$v_i(t) = v_f \tag{3}$$

We now want to derive a recursive algorithm to compute the constant acceleration of each vehicle while satisfying the following three constraints:

(i) v_f is known
(ii) $a_i > a_{i+1}$
(iii) each a_i is constant

We assume v_f is dynamic since it may change with road conditions or long-term technological changes (e.g. road and rubber technologies). But at a specific moment in time, v_f is a known constant and so is either a_1 or τ_1. This values can be experimentally determined by safety and efficiency criteria. Finally we get the following relation,

$$v_f = a_1\tau_1 \tag{4}$$

We also define the distance between two adjacent cars as $d_{i-1,i}$. For example, the distance between the first two cars is,

$$d_{1,2}(\tau_1) = s_1(\tau_1) - s_2(\tau_1) \tag{5}$$

Let ℓ be the optimal safe spacing determined by v_f (and therefore the road conditions). We want to solve for the optimal a_i of all n vehicles under the constraint that

$$d_{i-1,i}(\tau_i) = s_{i-1}(\tau_i) - s_i(\tau_i) = \ell \tag{6}$$

Under the conditions of Eq.s 1 - 6, we can solve for the algorithm that determines the acceleration of each car_i which is defined recursively as,

$$a_i = \frac{v_f^2}{2\ell + \frac{v_f^2}{a_{i-1}}} \tag{7}$$

Since we assume the parameter ℓ is related to factors that delay the breaking speed of the platoon, like communication lag and road conditions, ℓ will proportionally increase as speed increases. We let μ be the coefficient that represents the breaking delay and we get

$$\ell_i = \mu v_i(t) \tag{8}$$

Therefore, at the speed limit we get

$$\ell = \mu v_f \tag{9}$$

Since a_i is partially defined in terms of ℓ, it can be simplified to

$$\begin{aligned} a_i &= \frac{v_f^2}{2\mu v_f + \frac{v_f^2}{a_{i-1}}} \\ a_i &= \frac{v_f}{2\mu + \frac{v_f}{a_{i-1}}} \end{aligned} \tag{10}$$

Therefore, the acceleration of any car n in a platoon can be defined by

$$a_n(t) = \begin{cases} a_1(t) = \frac{v_f}{\tau_1}, & 0 < t < \tau_1 \\ a_i(t) = \frac{v_f}{2\mu + \frac{v_f}{a_{i-1}}}, & 0 < t < \tau_i \end{cases} \tag{11}$$

See Appendix for more details on the derivation of the acceleration algorithm.

The following is a complete recursive algorithm to find the optimal mean acceleration of each vehicle in the queued platoon:

Algorithm 1 Constant Mean Accelerations

```
v_f = speed limit
a_1 = acceleration of the first car
n = number of cars in platoon
for i in range(n) do
    a_i = v_f / (2μ + v_f / a_{i-1})
end for
```

B. Extension to a Natural Acceleration Profile

Since constant acceleration is very ideal, uncomfortable, and therefore impractical, we will modify our algorithm using Akcelick and Biggs's [19] polynomial acceleration model developed from experimentally-observed, acceleration patterns.

Akcelick and Biggs studied the acceleration profiles of vehicles under real-life traffic scenarios, concluding that the following functional representation of the acceleration profile provides the best fit to the observed real-life data. This particular acceleration profile permits the position and value of the maximum acceleration achieved to vary for a given average acceleration rate. The general formula, found by them, is:

$$a_i(t) = r a_m \theta^n (1 - \theta^m)^2 \tag{12}$$

where a_m is the maximum acceleration of car_i, θ is the time ratio t/τ_i,and n, r , and m are parameters that allow flexible acceleration profiles, derived from a fourth parameter, ρ.

We use the constant acceleration values derived from Eq. 11 to find the entire acceleration profile of each car in the queued platoon:

$$a_i(t) = 4.26 a_i \left(\frac{t}{\tau_i}\right)\left(1 - \left(\frac{t}{\tau_i}\right)^{3.2}\right)^2 \tag{13}$$

See Appendix for more details on the derivations of the parameter values 4.26 and 3.2.

Algorithm 2 shows the complete recursive algorithm to find the optimal acceleration profiles of each vehicle in the queued platoon:

Figure 2 is an illustrative plot of the acceleration for five platooning vehicles with parameters of $a_1 = 10\frac{\text{km}}{\text{hr}\cdot\text{s}}$, $v_f = 50\frac{\text{km}}{\text{hr}}$ and $\mu = .2\text{s}$

Algorithm 2 Natural Acceleration

$v_f =$ speed limit
$a_1 =$ average acceleration of the first car
$n =$ number of cars in platoon
for i in range(n) **do**
  $a_i = \dfrac{v_f}{2\mu + \frac{v_f}{a_{i-1}}}$ ▶ average acceleration of car_i
  $\tau_i = \dfrac{v}{a_i}$ ▶ time car_i is accelerating
  $a_i(t) = 4.26a_i(\frac{t}{t_i})(1-(\frac{t}{t_i})^{3.2})^2$
end for

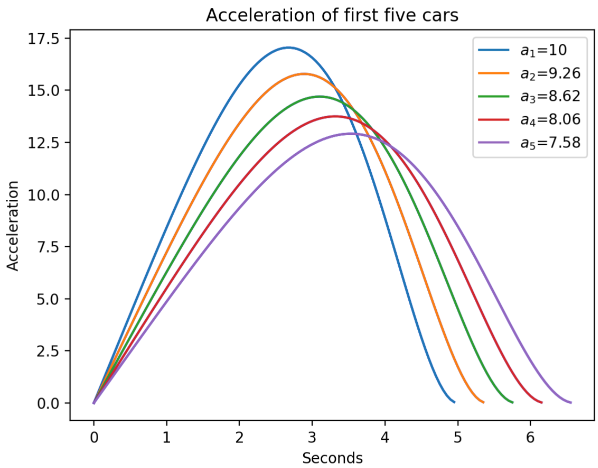

Fig. 2: Acceleration of the first five cars in a platoon with average acceleration of $10\frac{\text{km}}{\text{hr}\cdot\text{s}}$ for car_1 and $v_f = 50\frac{\text{km}}{\text{hr}}$

III. Analysis of Flow Rates from Traffic Stop

In this section we will first look at how we can develop equations describing the efficiency of our acceleration scheme. Afterwards, we will look at how different parameters affect flow rate.

A. Equations describing flow rate

To asses how efficient our scheme is, we will determine how many cars can get through the intersection. We define this as the amount of cars that reach the intersection during a green light cycle. For example, we assume the first car is through at $t = 0$ because it is at the beginning of the intersection. From this we see that the distance of the i-th car is proportional to the length of the cars and the distance between them. We assume the average car length is 25 feet [20]. Since the cars are semi-autonomous we can also assume constant distance between them. From this we get,

$$\mathcal{D}_i = (i-1)\lambda \tag{14}$$

where $\mathcal{D}_i$ is the distance from the intersection of the i'th vehicle and λ is the length of the vehicle plus the space in-between each stopped vehicle. Multiplying by (i-1) gives us the sum of the distance of every car up to the i'th car. Now we can determine how many cars reach the intersection, but first we need to acknowledge that two cases are possible:

(i) the last car to reach the intersection during the green light is traveling at v_f upon arriving at the intersection, or,
(ii) the last car to reach the intersection is traveling at $v_i(t) < v_f$ (it is still accelerating.)

Under the first case, where the last car to reach the intersection has already reached v_f, we have the following closed form solution for the number of cars to pass through during a green light cycle (derived in Appendix A):

$$i = \frac{v_f(t_d - \frac{v_f}{2a_1})}{\lambda + v_f\mu} + 1 \tag{15}$$

Under the latter case, where the last vehicle has yet to reach final velocity, we must numerically solve for the number of cars to pass through the intersection using the following result derived in Appendix B:

$$\frac{4.26a_i}{t_i}\left(\frac{t^3}{6} - \frac{2}{32.34t_i^{3.2}}t^{6.2} + \frac{t^{9.4}}{78.96t_i^{6.4}}\right) = (i-1)\lambda \tag{16}$$

B. Analysis of flow rate

We can now analyze the efficiency of our algorithm with different values of v_f, a_1 and μ. We plot the number of cars through the intersection with respect to time. Each point except the last one represents a car that has not reached v_f at the intersection. The last point represents the amount of cars through in a given period of time. This convention facilitates the analyses by (i) eliminating redundancy since the amount of cars reaching the intersection at speed, v_f, increases at a constant rate and, (ii) clearly showing when the cars have reached v_f at the intersection.

Figure 3 shows that as μ decreases, the time it takes each car to reach the intersection decreases. This is due to the nature of the acceleration scheme where a larger μ means a smaller acceleration for every car. Therefore, as μ approaches zero, the amount of cars through will approach a straight line where $a_i = a_1$. As a reminder, μ is the coefficient used to determine safe spacing between cars at a given velocity.

Following the opposite trend, and keeping everything else constant, as a_1 increases, the time it takes each subsequent

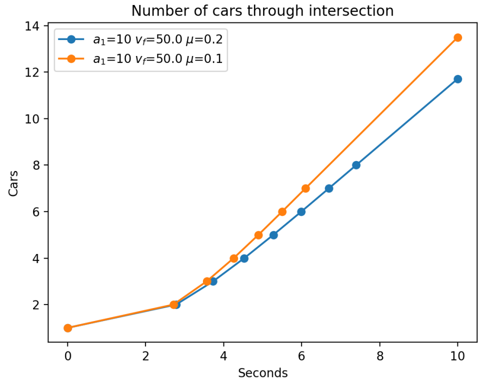

Fig. 3: Varying μ; a dot represents a car passing through the intersection at a velocity $< v_f$.

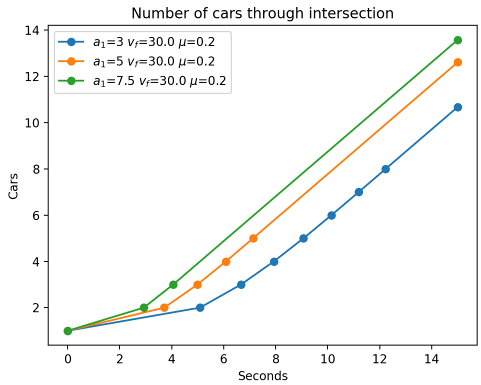

Fig. 4: Varying a_1; a dot represents a car passing through the intersection at a velocity $< v_f$.

car to reach the intersection decreases. Additionally, the amount of cars reaching the intersection at speed v_f increases. For example, Figure 4 shows how at $a_1 = 7.5$ only 3 cars have not reached v_f at the intersection while, at $a_1 = 3$, many cars pass the intersection without reaching v_f.

Oddly enough, when only v_f changes, we see that for a short period of time a smaller v_f is more efficient, ie. a small number of cars are in the platoon. Figure 5 demonstrates that when $a_1 = 5$, cars 2 to 5 pass through the intersection sooner when the speed limit is smaller. This phenomenon is illustrated more clearly in Figure 6 where we see that accelerations for a smaller v_f are quicker for the first five cars which explains why they pass the intersection sooner.

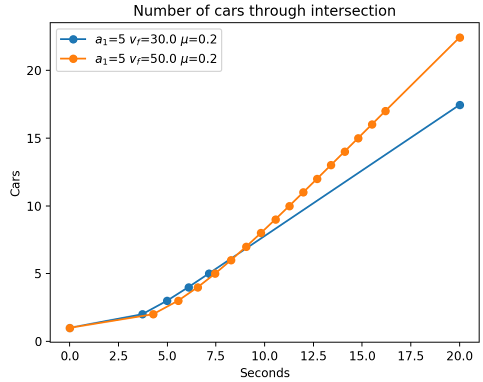

Fig. 5: Varying v_f; a dot represents a car passing through the intersection at a velocity $< v_f$.

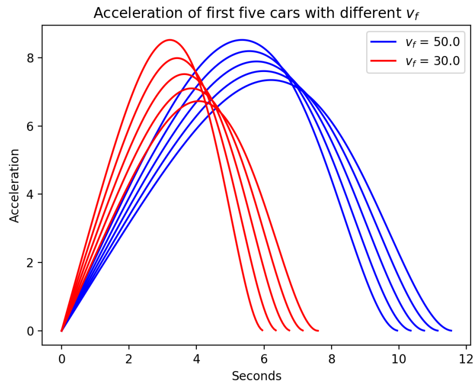

Fig. 6: The first five vehicles to pass through an intersection with different v_f.

IV. Discussion of Implementation under Current Infrastructure

IEEE has developed the 802.11p standard, for vehicular communication, to allow information exchange between vehicles in a V2V fashion[25]. The development of the AV software systems has even enabled for a vehicle's state awareness to extend beyond its immediate sensor readings [13]. These same communication protocols may be extended to Vehicle to Infrastructure (V2I) communications, where adoption has already been implemented into DRSC devices with the intent of broadcasting traffic light information for intersection handling. The acceleration algorithm presented easily integrates under any of these scenarios, whether under currently-driven semi-autonomous vehicles, such as Tesla's Autopilot [4] or Google's Waymo cars [1], or tomorrow's more advanced V2I smart intersections and traffic management.

First presented in a 2005 paper for AVs, titled "Multi-agent Traffic Management: An Improved Intersection Control Mechanism," a system to synchronize autonomous cars with traffic lights was presented. Their design consisted of cars reserving passage through intersections so that vehicles did not have to stop [21]. This idea has since been extended by MIT researches to a slot-based intersection for continuous traffic [22]. Found to significantly reduce delay and double intersection capacity, it is arguably a path forward for traffic control systems engaging in a centralized vehicle coordination system. Another intersection strategy to reduce inefficiencies was to coordinate the behavior of multiple traffic lights in a ”Smart Traffic Light Control System” [23]. This research focused on synchronizing traffic lights to reduce the amount of stops for each car.

While these systems would greatly decrease inefficiencies, our algorithm helps enable their implementation by integrating a missing component seamlessly into them: adding an optimal acceleration algorithm and pattern for each of the AVs in the system to utilize. One of the strongest advantages to our acceleration algorithm is that it is easily implemented today, since it only requires a slight modification of the vehicle's software layer. Cars are constantly being redesigned and, as the nascent autonomous vehicle industry develops, it will be easy for any and all car manufacturers to implement this acceleration algorithm by simply including it in a vehicle's software suite.

Only three requirements are necessary for the implementation of this algorithm today. First is software: car manufacturers will have to develop programs that find the optimal acceleration of the i'th vehicle. This is easily implemented using the described algorithm above. Next is hardware. Since most vehicles currently come installed with all the actuators necessary to perform software-controlled acceleration, steering, and braking, additional hardware may only be necessary to perform oversight that cars are accelerating properly or maintaining safe distances (ie. sensors to measure distance between cars). The V2V, or V2I, hardware components detailed in [15] are clearly a necessary condition for implementation. Lastly, “awareness” of non-autonomous vehicles is essential. Our algorithm presented can only coordinate between cooperating vehicles.

While not detailed in this paper, current work is being undertaken to describe the cooperation between AVs utilizing this acceleration algorithm in the presence of non-autonomous vehicles. According to SAE's levels of autonomy, our algorithm only requires a level 1 autonomy where acceleration is assisted but as cars transition upward within SAE's levels of autonomy, platooning will be easier to implement and vehicles should easily know when there are non-autonomous vehicles present.

V. Concluding Remarks

Vehicular communication has enabled cooperation between multiple autonomous vehicles (AVs) and, in particular, state estimation from multiple sources of information coming from multiple vehicles. Our work presents a concise -and optimal- piece of the puzzle, with the sharing of future trajectories, to coordinate the motion of acceleration to make navigation safer and more efficient for semi-autonomous vehicles.

Our algorithm offers a simple solution to alleviate road congestion through higher throughput and to improve road safety by removing human error from vehicle operation by providing an algorithm to compute, in real-time, the optimal acceleration of a group of semi-autonomous vehicles, communicating and acting in a cooperative, coordinated manner. Our algorithm enables a group of vehicles to maximize throughput at any position in the roadway where vehicles find themselves accelerating and, additionally, explicitly accounts for any desired level of safety, dynamically adjusts for changes in road conditions and future technological changes (ie. rubber tires and asphalt surfaces), and can be implemented using simple vehicle connectivity and communication.

We have shown that our acceleration algorithm increases efficiency of intersection throughput without sacrificing safety and under little change to the current road infrastructure, enabling it to be implemented as soon as possible. It also serves as a building block for any coordinated control systems, whether centralized, slot-based traffic control systems, or otherwise. Additionally, due to the construction of the algorithms, we believe it will be easy to implement under current vehicle technologies using V2V in semi-autonomous states, as well as easily integrating into more complex research proposals that require changes to the current roadway infrastructure and advanced V2I systems. Our algorithm provides a strong basis for individual acceleration in groups of semi- and fully-autonomous vehicles for the foreseeable future.

Acknowledgements

Acknowledgements omitted for blind review.

References

[1] Google Self-Driving Car Project. Available online: https://www.google.com/selfdrivingcar/ (accessed on 2 February 2019).

[2] Pittsburgh, Your Self-Driving Uber is Arriving Now. Available online: https://newsroom.uber.com/ pittsburgh-self-driving-uber/ (accessed on 30 December 2019).

[3] Newton, C. Uber will eventually replace all its drivers with self-driving cars. Verge 2014, 5, 2014.

[4] Tesla Motors: Model S Press Kit. Available online: https://www.tesla.com/presskit/autopilot (accessed on 30 December 2016).

[5] Inrix.com

[6] Schrank, D.; Eisele, B.; Lomax, T.; Bak, J. Urban Mobility Scorecard; Texas A&M Transportation Institute: College Station, TX, USA, 2015.

[7] Dunn Engineering Asociates, "Traffic Control System Handbook," *Federal Highway Administration Office of Transportation Management*, Washington, D.C., 2005, pp.8.

[8] Levy, J.I.; Buonocore, J.J.; Von Stackelberg, K. Evaluation of the public health impacts of traffic congestion: A health risk assessment. Environ. Health 2010, 9, 65.

[9] Systematics, C. Crashes vs. Congestion–What's the Cost to Society? American Automobile Association: Chicago, IL, USA, 2011.

[10] LaValle, S.M.; Hutchinson, S.A. Optimal motion planning for multiple robots having independent goals. IEEE Trans. Robot. Autom. 1998, 14, 912–925.

[11] Van Den Berg, J.P.; Overmars, M.H. Prioritized motion planning for multiple robots. In Proceedings of the 2005 IEEE/RSJ International Conference on Intelligent Robots and Systems, Edmonton, AB, Canada, 2–6 August 2005; pp. 430–435.

[12] Velagapudi, P.; Sycara, K.; Scerri, P. Decentralized prioritized planning in large multirobot teams. In Proceedings of the 2010 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), Taipei, Taiwan, 18–22 October 2010; pp. 4603–4609.

[13] Kim, S.W.; Chong, Z.J.; Qin, B.; Shen, X.; Cheng, Z.; Liu, W.; Ang, M.H. Cooperative perception for autonomous vehicle control on the road: Motivation and experimental results. In Proceedings of the 2013 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), Tokyo, Japan, 3–7 November 2013; pp. 5059–5066.

[14] Kim, S.W.; Qin, B.; Chong, Z.J.; Shen, X.; Liu,W.; Ang, M.H.; Frazzoli, E.; Rus, D. Multivehicle Cooperative Driving Using Cooperative Perception: Design and Experimental Validation. IEEE Trans. Intell. Transp. Syst. 2015, 16, 663–680.

[15] Shen, X.; Chong, Z.; Pendleton, S.; James Fu, G.; Qin, B.; Frazzoli, E.; Ang, M.H., Jr. Teleoperation of On-Road Vehicles via Immersive Telepresence Using Off-the-shelf Components. In Intelligent Autonomous Systems 13; Advances in Intelligent Systems and Computing; Springer International Publishing: Berlin, Germany, 2016; Volume 302, pp. 1419–1433.

[16] Switkes, J.P.; Gerdes, J.C.; Berdichevsky, G.; Berdichevsky, E. Systems and Methods for Semi-Autonomous Vehicular Convoys. U.S. Patent 8,744,666, 3 June 2014.

[17] ATA Technology and Maintenance Council, "White Paper: Automated Driving and Platooning Issues and Opportunities," *Automated Driving and Platooning Task Force*, 2015.

[18] Bergenhem, Carl, "Overview of Platooning Systems," *19th ITS World Congress*, Vienna, Austria, 2012.

[19] Akcelik, R. and Biggs, D.C., "Acceleration profile models for vehicles in road traffic," *Transportation Science, 21*, 1987, pp 36-54.

[20] New England Section ITE Technical Committee, "EVALUATION OF AVERAGE EFFECTIVE VEHICLE LENGTH IN QUEUE," *http://neite.org/Documents/Technical/ITE%20Tech%20Comm%20Queue%20Study-Final.pdf.*

[21] Dresner, Kurt, "Multiagent Traffic Management: An Improved Intersection Control Mechanism," *Department of Computer Sciences*, Austin, TX, 2005.

[22] Tachet R, Santi P, Sobolevsky S, Reyes-Castro LI, Frazzoli E, Helbing D, et al. (2016) "Revisiting Street Intersections Using Slot-Based Systems." Tieqiao Tang, Beihang University, CHINA, 2016.

[23] Ghazal, Bilal, "Smart Traffic Light Control System," *IEEE*, 2016.

[24] Shen, X.; Chong, Z.J.; Pendleton, S.; Liu,W.; Qin, B.; Fu, G.M.J.; Ang, M.H. Multi-vehicle motion coordination using v2v communication. In Proceedings of the 2015 IEEE Intelligent Vehicles Symposium (IV), Seoul, Korea, 28 June–1 July 2015; pp. 1334–1341.

[25] Fernando A. Teixeira, Vinicius F. e Silva, Jesse L. Leoni, Daniel F. Macedo, and José M.S. Nogueira. 2014. Vehicular networks using the IEEE 802.11p standard. Veh. Commun. 1, 2 (April 2014), 91–96. DOI:https://doi.org/10.1016/j.vehcom.2014.04.001

Appendix

From $t = 0$ to $t = t_{i-1}$, car $i-1$ is accelerating. After $t = t_{i-1}$ the car maintains a constant velocity, v_f, so $s_{i-1}(t_i)$ can be written as

$$s_{i-1}(t_i) = \frac{1}{2}a_{i-1}t_{i-1}^2 + v_f(t_i - t_{i-1}) \tag{17}$$

We now expand $s_{i-1}(t_i)$ and $s_i(t_i)$

$$\ell = s_{i-1}(t_i) - s_i(t_i) = \frac{1}{2}a_{i-1}t_{i-1}^2 + v_f(t_i - t_{i-1}) - \frac{1}{2}a_i t_i^2 \tag{18}$$

From this equation we can see that a_i can be defined as a function of ℓ, t, and a_{i-1}. So to solve for a_i we begin by replacing t_i and t_{i-1} with $\frac{v_f}{a_i}$ and $\frac{v_f}{a_{i-1}}$ respectively.

$$\ell = \frac{1}{2}\frac{v_f^2}{a_{i-1}} + v_f(\frac{v_f}{a_i} + \frac{v_f}{a_{i-1}}) - \frac{1}{2}\frac{v_f^2}{a_i} \tag{19}$$

and now everything can be reduced to

$$\ell = \frac{1}{2}v_f^2(\frac{1}{a_i} - \frac{1}{a_{i-1}}) \tag{20}$$

finally, we solve for a_i

$$\begin{aligned} 2\ell &= \frac{v_f^2}{a_i} - \frac{v_f^2}{a_{i-1}} \\ 2\ell + \frac{v_f^2}{a_{i-1}} &= \frac{v_f^2}{a_i} \\ a_i &= \frac{v_f^2}{2\ell + \frac{v_f^2}{a_{i-1}}} \end{aligned} \tag{21}$$

Akcelick and Biggs [19] defined the parameter ρ as the following:

$$\rho = \frac{2m^2 + 15m + 19}{3(m+3)(2m+3)}$$

Akcelick and Briggs derived their function by randomly following cars and recording the values at intersections. They found that setting n to 1 works well and reduces complexity. They also define ρ as a function of the average, initial and final velocity of the car

$$\rho = \frac{\bar{v} - v_i}{v_f - v_i} \tag{22}$$

where

$$\bar{v} = x_i/t_i x_i = \frac{a_i t_i^2}{2} \tag{23}$$

From this we see that $\bar{v}$ is the average velocity and v_i is the initial velocity. Since the cars are accelerating from stop, v_i is equal to zero which means that ρ is constant for every car:

$$\rho = \frac{x_i}{t_i}\frac{1}{v_f}$$

t_i cancel and replace $a_i t_i$ with v_f,

$$\begin{aligned} \rho &= \frac{a_i t_i^2}{2t_i}\frac{1}{v_f} \\ \rho &= \frac{a_i t_i}{2v_f} = \frac{v_f}{2v_f} = \frac{1}{2} \end{aligned} \tag{24}$$

Now we can solve for m which is also constant for every car

$$\begin{aligned} \frac{1}{2} &= \frac{2m^2 + 15m + 19}{3(m+3)(2m+3)} \\ \frac{3}{2}2m^2 + 9m + 9 &= 2m^2 + 15m + 19 \\ m^2 - 1.5m - 5.5 &= 0 \\ m &= \frac{-(-1.5) + \sqrt{(-1.5)^2 - 4(-5.5)(1)}}{2(1)} \\ m &\approx 3.2 \end{aligned} \tag{25}$$

All that's left is finding the values of ra_m which can be attained by the following equation:

$$\begin{aligned} ra_m &= \frac{2\bar{a}(m+1)(m+2)}{m^2} \\ &= 4.26\bar{a} \end{aligned} \tag{26}$$

where $\bar{a}$ represents the average acceleration,

$$\bar{a} = \frac{v_f - v_i}{t_i}, \tag{27}$$

but since $v_i = 0$, we get

$$\bar{a} = \frac{v_f}{t_i} = a_i, \tag{28}$$

So we can finally define the acceleration of every car as

$$a_n(t) = 4.26a_i(\frac{t}{t_i})(1 - (\frac{t}{t_i})^{3.2})^2 \tag{29}$$

In this section we detail the analytical derivations of the number of cars that pass through the intersection.

A. Final Car is travelling v_f upon arrival

First we will look at the case where the last car has finished accelerating. We want to know how long it takes a car, indicated by t_d, to reach the intersection after reaching v_f, therefore we know it has already traveled a certain distanced indicated by,

$$s_i(t_i) = \frac{a_i t_i^2}{2} = \frac{v_f^2}{2a_i} \tag{30}$$

so we want to solve for t_d by finding the distance that is left, giving us

$$v_f(t_d - \frac{v_f}{a_i}) = (i-1)\lambda - \frac{v_f^2}{2a_i} \tag{31}$$

It may be apparent that at v_f the cars are traveling equal distances apart, but not as apparent, the number of cars through the intersection increases at a constant rate. To show this we first solve for t_d

$$\begin{aligned} v_f(t_d - \frac{v_f}{a_i}) &= (i-1)\lambda - \frac{v_f^2}{2a_i} \\ t_d - \frac{v_f}{a_i} &= \frac{(i-1)\lambda - \frac{v_f^2}{2a_i}}{v_f} \\ t_d &= \frac{(i-1)\lambda}{v_f} - \frac{v_f^2}{2a_i v_f} + \frac{v_f}{a_i} \\ t_d &= \frac{(i-1)\lambda}{v_f} + \frac{v_f}{2a_i} \end{aligned} \tag{32}$$

Since it is a recursive equation we can also represent t_d in terms of a_{i-1}:

$$\begin{aligned} t_d &= \frac{(i-1)\lambda}{v_f} + \frac{v_f}{\frac{2v_f}{2\mu + \frac{v_f}{a_{i-1}}}} \\ t_d &= \frac{(i-1)\lambda}{v_f} + \mu + \frac{v_f}{2a_{i-1}} \end{aligned} \tag{33}$$

From both iterations of t_d, we can show that the change in t_d of every car is constant,

$$\begin{aligned} t_d(i) = \frac{(i-1)\lambda}{v_f} + \frac{v_f}{2a_i} &= \frac{(i-1)\lambda}{v_f} + \mu + \frac{v_f}{2a_{i-1}} \\ \Delta t_d &= t_n(i) - t_n(i-1) \\ \Delta t_d = \frac{(i-1)\lambda}{v_f} + \mu + \frac{v_f}{2a_{i-1}} &- (\frac{(i-2)\lambda}{v_f} + \frac{v_f}{2a_{i-1}} \\ \Delta t_d &= \frac{\lambda}{v_f} + \mu \end{aligned} \tag{34}$$

We now can see that the amount of cars through the intersection is dependent on the distance between cars, the final speed and "delay" coefficient. But we can also solve for the amount of cars through given a specific time window. The recursive nature of the acceleration scheme means we can define t_d with a_1, we know that going from a_i to a_{i-1} add μ so we see that

$$t_d = \frac{(i-1)\lambda}{v_f} + (i-1)\mu + \frac{v_f}{2a_1} \tag{35}$$

with a little bit of algebra we can solve for i,

$$\begin{aligned} t_d &= \frac{(i-1)\lambda}{v_f} + (i-1)\mu + \frac{v_f}{2a_1} \\ \frac{(i-1)\lambda}{v_f} + (i-1)\mu &= t_d - \frac{v_f}{2a_1} \\ i - 1 &= \frac{v_f(t_d - \frac{v_f}{2a_1})}{\lambda + v_f\mu} \\ i &= \frac{v_f(t_d - \frac{v_f}{2a_1})}{\lambda + v_f\mu} + 1 \end{aligned} \tag{36}$$

B. Final Car is travelling $< v_f$ upon arrival

We have finished deriving equations describing the efficiency of the acceleration scheme for long time intervals where the cars have finished accelerating by the time they reach the intersection. Now we can begin to look at how many cars get through before they are done accelerating. To do so, we need to integrate the polynomial equation to find velocity and distance traveled

$$
\begin{aligned}
v_i(t) &= \int 4.26a_i(\frac{t}{t_i})(1-(\frac{t}{t_i})^{3.2})^2 \\
&= \int 4.26a_i(\frac{t}{t_i})(1-\frac{2}{t_i^{3.2}}t^{3.2}+\frac{t^{6.4}}{t_i^{6.4}}) \\
&= \int \frac{4.26a_i}{t_i}(t-\frac{2}{t_i^{3.2}}t^{4.2}+\frac{t^{7.4}}{t_i^{6.4}}) \\
&= \frac{4.26a_i}{t_i}(\frac{t^2}{2}-\frac{2}{5.2t_i^{3.2}}t^{5.2}+\frac{t^{8.4}}{8.4t_i^{6.4}}) \\
x_i(t) &= \int \frac{4.26a_i}{t_i}(\frac{t^2}{2}-\frac{2}{5.2t_i^{3.2}}t^{5.2}+\frac{t^{8.4}}{8.4t_i^{6.4}}) \\
&= \frac{4.26a_i}{t_i}(\frac{t^3}{6}-\frac{2}{32.34t_i^{3.2}}t^{6.2}+\frac{t^{9.4}}{78.96t_i^{6.4}})
\end{aligned} \tag{37}
$$

To solve for time we now have

$$
\frac{4.26a_i}{t_i}(\frac{t^3}{6}-\frac{2}{32.34t_i^{3.2}}t^{6.2}+\frac{t^{9.4}}{78.96t_i^{6.4}})=(i-1)\lambda \tag{38}
$$

We can't simplify for t so this equation has to be solved numerically. We can graph the values of t_d for every car_i to visualize the efficiency of the scheme.