Cooperative Intersection Automation Using Virtual Platoons

Alejandro Ivan Morales Medina*, Nathan van de Wouw* and Henk Nijmeijer*
**Department of Mechanical Engineering, Eindhoven University of Technology, the Netherlands*

Summary. Cooperative Intersection Control achieves a safe and efficient flow of vehicles through an intersection while taking into account the dynamics of the individual vehicles. This objective is achieved using the concept of virtual platooning.

Cooperative Intersection Control

The Cooperative Intersection Control (CIC) aims to automate a road intersection to achieve an efficient, safe and smooth flow of vehicles through the intersection. This objective is achieved using the virtual platooning concept. Virtual platooning allows to form platoons of vehicles traveling on different lanes of the intersection with different directional intentions. Existing intersection management approaches focus on the high-level scheduling of the crossing sequence and tend to ignore realistic vehicle dynamics aspects. In contrast, CIC considers both low- and high-levels of execution and automation. A preliminary version of CIC for a simple intersection was presented in [1]. Here, an approach for generic intersections is proposed.

System Architecture

The architecture of CIC is depicted in Figure 1. It is divided in an execution level and a supervision level. The execution level ensures that the relative motion between vehicles is such that every vehicle that crosses the intersection is at a small but safe distance from other vehicles, while maintaining a constant velocity. The supervision level forms the virtual platoon and decides which control mode, for every vehicle, is activated for virtual platooning. The intersection geometry is generalized to accommodate to generic intersection layouts. For instance, Figure 2 depicts a four-way intersection with several possible trajectories.

Execution Level Control

Both the lateral and longitudinal motions of every vehicle are controlled.

Lateral control

Every vehicle is modeled using the kinematic car-like model, presented in [2], referenced to a Frénet frame, which defines the orthogonal projection of a point in space to a given curve. The resulting coordinates are, namely, the distance from the point to the curve, the traveled distance along the curve (or path coordinate), the orientation error (defined as the difference between the orientation of the velocity vector of the vehicle and the tangent angle at the path coordinate), and the curvature at the path coordinate. These set of coordinates is then mapped into the chained form coordinate space, also presented in [2]. In this state space, a time-varying linear feedback law is designed to achieve the regulation to zero of the distance from the point to the curve and the orientation error. In this way, path following control is achieved to support automated steering..

Longitudinal control

Three control modes are designed for longitudinal automation, namely, Cruise Control (CC), which is a velocity controller, Cooperative Adaptive Cruise Control (CACC), which is a inter-vehicle distance controller [3], and Virtual CACC

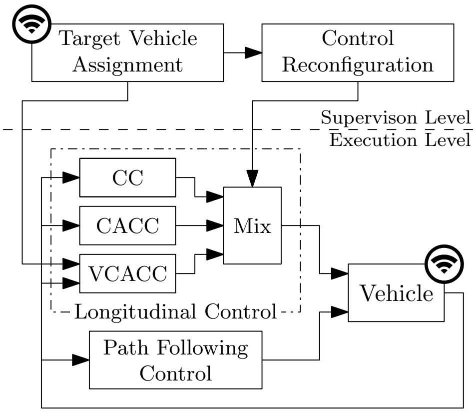

Figure 1: System architecture.

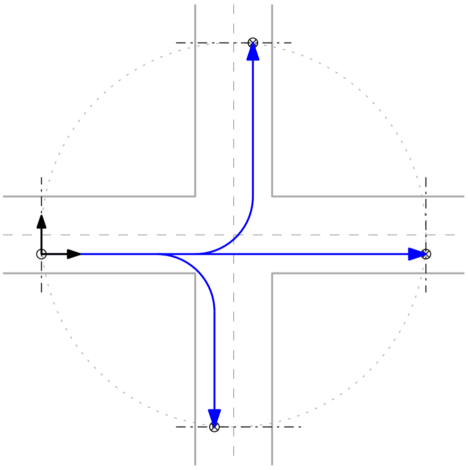

Figure 2: Intersection with four lanes, and several possible trajectories.

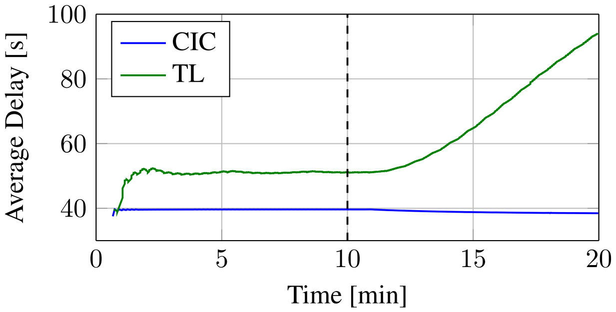

Figure 3: Comparison of the average delay in the intersection for a sudden change in flow.

(VCACC), which is a CACC generalization that allows to control a *virtual* inter-vehicle distance. The control effort of each controller is then mixed by calculating a convex combination of the individual control efforts that achieves a smooth control reconfiguration, as described in [4].

Each vehicle drives a distinct trajectory through the intersection, which is defined based on its entry and exit lane. A pair of trajectories can be either non-crossing or crossing. If the trajectories are non-crossing the vehicles can cross the intersection without any conflict. On the other hand, if the trajectories are crossing, the relative motion between vehicles has to avoid the so-called collision point of the trajectories. The collision point lays on both crossing trajectories but not necessarily at the same distance to collision, which is the distance, along a trajectory, measured from the entry point of a certain lane to the collision point. Therefore, the virtual inter-vehicle distance between a target vehicle V_n and a host (i.e. following) vehicle V_m is defined as

$$\delta = s_n - s_m - S_n + S_m - L_m \tag{1}$$

where s_i, for $i \in \{n, m\}$, is the path coordinates along a trajectory, S_i, for $i \in \{n, m\}$, is the distance to collision, and L_m is the length of the vehicle. Note that the target vehicle V_n will always cross the intersection first followed by the host vehicle V_m. This virtual inter-vehicle distance can be used by the CACC to regulate a virtual reference inter-vehicle distance, thereby achieving the aforementioned safe relative motion between vehicles.

Supervision Level

The supervisory automation level involves the target vehicle assignment to every vehicle and executes the control reconfiguration, see Figure 2. When a vehicle enters the intersection region the supervisory controller checks whether there is a vehicle in the intersection with which it has a crossing trajectory. If such vehicle exists, the control mode is reconfigured to VCACC, if not, the control mode is assigned to CC if there is a free road in front of the vehicle, or to CACC if there is a vehicle in front. The target vehicle assignment is based on a First-Come-First-Serve intersection access protocol. Once a vehicle has crossed the intersection (meaning $s_m \geq S_m$, the control mode is assigned to CC or CACC based on the same aforementioned conditions.

Results

The CIC has proven to yield a higher throughput and a lower average delay (which is defined as the time a vehicle spends inside the intersection region) compared to a traffic light controlled intersection. The comparison is made using straight trajectories only, constant vehicular flow, and fixed traffic light schedule. An interesting result is the immediate response of the CIC to a sudden change in the vehicular flow without prior knowledge of the intensity of the flow. Figure 3 shows that the average delay remains almost constant after the change in vehicular flow (depicted by the vertical dashed line), in contrast to the average delay for the traffic light (TL) scenario.

Conclusion

The Cooperative Intersection Control methodology automates a road intersection using the virtual platooning concept. This methodology proves to be robust to the sudden changes in vehicular flow.

References

[1] A.I. Morales Medina, N. van de Wouw, and H. Nijmeijer. Automation of a t-intersection using virtual platoons of cooperative autonomous vehicles. In *Intelligent Transportation Systems (ITSC), 2015 IEEE 18th International Conference on*, pages 1696–1701, 2015.

[2] P. Morin and C. Samson. *Handbook of Robotics*, chapter Motion control of wheeled mobile robots. Springer-Verlag Berlin Heidelberg, 2008.

[3] J. Ploeg, B.T.M. Scheepers, E. van Nunen, N. van de Wouw, and H. Nijmeijer. Design and experimental evaluation of cooperative adaptive cruise control. In *Intelligent Transportation Systems (ITSC), 14th International IEEE Conference on*, pages 260–265, 2011.

[4] E. van Nunen, A.I. Morales Medina, and H. Nijmeijer. Control reconfiguration for cooperative adaptive cruise control ensuring safety and comfort. In *FISITA World Automotive Congress 2014*, 2014.