13th ITS European Congress, Brainport, the Netherlands, 3-6 June 2019

Paper number ITS-XXXX

A Cooperative Intersection Support Application enabled by SAFE STRIP Technology both for C-ITS equipped, non-equipped and autonomous vehicles

Giammarco Valenti[1*], Davide Piscini[1], Francesco Biral[1]

1. Department of Industrial Engineering, University of Trento, Italy

mail: *{giammarco.valenti}@unitn.it

Abstract

This paper extends a cooperative intersection support system which exploits the concept of mirroring in cognitive science to perform inference of intention and here boosted with additional and more accurate information on intersection scenario provided by the infrastructure. Design principles and the implementation of the algorithms are presented. Several novelties are introduced, among which path planning with human-like trajectories, multiple vehicles intersections and the right of way management. The agent introduced is scalable and general and can be used to generate the manoeuvres for autonomous agents and warnings within driver assistance systems. Simulation results are presented in both those frameworks, autonomous intersection management and driver assistance scenario (warning system).

Keywords:

Advanced driver assistance systems, intersection support, intersection control, driver modelling.

Introduction

This paper extends a warning intersection support system [1] enabled by SAFE STRIP technology, which allows to detect and get data of C-ITS equipped, non-equipped vehicles by means of infrastructure based sensors [26].

A great portion of all car accidents involves intersections and many of them are caused by drivers' lack of attention or the lack of understanding, resulting in taking dangerous actions. In recent reviews on support systems are [2][3][4] most of applications presented are based on optimization approaches where the first goal is safety and hierarchically other criteria are optimised such as efficiency, traffic throughput, energy efficiency, comfort, and delays. Some of the approaches are centralized [5][6][7] or decentralized [8], all of them requires exchange of information, using V2X and on-board sensors or sensors on the infrastructure: for example in [9] position and velocity data from fixed road sensors without considering the driver behaviour. More driver-centred approaches are based on classification of intentions by means of pattern recognition [10] [11]. Machine learning methods are used such as Support Vector Machine and Hidden Markov Models. A review of methods for classification of incoming vehicles is provided in [12]. The Intersection Support Application proposed here extends the one presented in [1], taking advantage of data provided by SAFE STRIP infrastructure [29]. Human like motion primitives are used to explore feasible goal-directed actions or manoeuvres here called the affordances [13]. Smooth path for predicting the motion of other road users, multiple vehicles and right of way rules are considered in this work as extensions of [1]. Finally the bio-inspired concept of "mirroring" is used to evaluate and discover the human intention [14][15][16][17]. The intention is then discovered comparing those affordances with the agent behaviour. Beyond the warning-based support, the paper also provides the basis to build a full autonomous intersection management, which is modular and scalable and powered by the ability to plan human-like manoeuvres which should improve user acceptance. The paper will demonstrate the working principle of the application and prove it efficacy with simulation results. The results are still preliminary and experimental analysis are planned in the future in the framework of SAFE STRIP.

Use case: application scenario

The application scenario considered is illustrated in Figure 1 (left). The scenario consists in a T-intersection with six connections between incoming and outgoing lanes (coloured arrows in Figure 1). In the considered scenario three vehicles approach the intersection and for each vehicle all the possible choices of connections are considered. This scenario adopted here is not a limitation but just representative of an intersection with all the elements to explain the concept and prove the system performance in simulations. Nevertheless, the method is callable to more complex scenario with different geometric layout and with more vehicles.

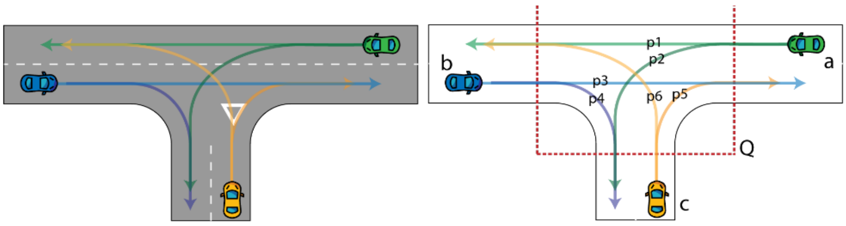

Figure 1 Use case: intersection scenario

The three vehicles, are labelled *a, b, c* respectively, see Figure 1 (right), one for all incoming lanes. For each vehicle all the possible paths (i.e. according driving rules) are considered, and they are labelled with p (path) and numbered from *1* to *6*. The red square represented by capital letter *Q* is the representation of the conflict area. This area cannot be occupied by two or more vehicles if they have intersecting trajectories, the square dimensions depend on the road geometry and on the overall feasible trajectories hypotheses.

Proposed architecture

The Architecture of the intersection support system is based on the work proposed in [1] (the codriver), which is the baseline of this paper. The main goal of this work is to modify that system and develop a new one able to deal with complex intersections and take full advantage of SAFE STRIP LDM. The action priming block shown in the following section is indeed the same as the one is the baseline work [1]. The system has to mirror the driver intentions in terms of danger, and provide warnings for hazardous behaviours, it has also to provide safe manoeuvres to follow for autonomous (or for intervention in) approaching to the intersection.

Intersection support application structure

The application structure is composed of two hierarchical layers: the top layer aims to solve the intersections in terms of available time to pass for each vehicle, and the second is required to the generation of human like longitudinal behavior compatible with the time slots and their evaluation
The layerer control architecture is used to model the human decision making of manoeuvres, making use of the longitudinal motion primitives [18][19], which are atomic longitudinal human-like maneuvers. The motion primitives are parameterized solutions of optimal control problems that minimize a cost function related to jerk and time. Those solutions resembles longitudinal maneuvers of human drivers and for this reasons are qualified as "human like [18]". The block representation of the selected architecture is shown in Figure 2. The driver selects one maneuver by means of the action selection block, but in our mirroring approach all the possible maneuvers are taken into consideration to identify which one the other users may use.

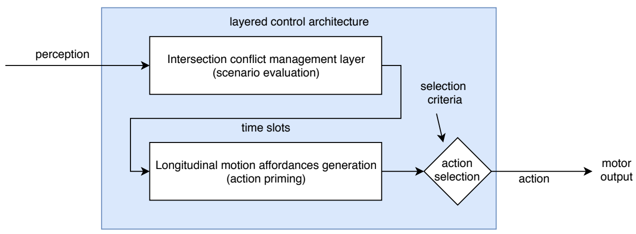

Figure 2 Hierarchical architecture used to model the human sensorimotor behaviour near the intersection

The model in Figure 2 is aimed to represent a driver which generates manoeuvres (or affordances) in the action priming block to pass the conflict area in the time slots provided by the higher layer. The aim is to assess the existence of safe manoeuvres within the possible ones rather than discovering the exact manoeuvre the driver intends to pursue.

Action priming: The action priming is the computation of the sets of possible longitudinal manoeuvres to approach the intersection, those manoeuvres can be of two types:

- *Stop manoeuvres*: The goal of this manoeuvres is to stop before the conflict area. Acceleration, velocity and distance from the intersection are required, (a_0, v_0, s_f respectively).

$$m_{stop} = stop(a_0, v_0, s_f)$$

- *Pass manoeuvres*: The goal of this manoeuvres is to reach the intersection (the conflict area Q) at a given time T. Those manoeuvres are constrained in time since the possibility to cross the intersection is given in terms of time slots, final velocity is not constrained.

$$m_{pass} = pass(a_0, v_0, s_f, T)$$

This type of manoeuvres represents single ones, in the process of action priming the objective is to generate the *set* of all possible manoeuvres given the time slots. The set of the stop manoeuvres is generated including all the possible ones that have final position shorter than s_f. The set of the pass manoeuvres can be instead represented by all the primitives with final time in the time slots, condition expressed by $T \in \mathcal{T}$, where $\mathcal{T}$ is the set of the time slots and it is a union of compact subsets. For example, $\mathcal{T} = [T_1, T_2] \cup [T_3, T_4]$ represents the possibility to pass in one of the two intervals, all the times are expressed from the current instant. The set of manoeuvres is:

$$M_{stop} = stop\left(a_0, v_0, s_f > s_{f0}\right)$$
$$M_{pass} = pass(a_0, v_0, T \in \mathcal{T})$$

Since $\mathcal{T}$ is a union of compact sets, the manoeuvres to compute are the extremal ones, two for each compact set. An example taken from our use case is shown in Figure 3, where a time slot is available, and both sets of manoeuvres are instantiated.

Mirroring of intention: The purpose of the system is to generate warnings for the driver. To accomplish the task, a comparison between the set of the safe manoeuvres and the driver behaviour is performed. First of all the sets of manoeuvres are represented in a *metric space*, this concept is borrowed by biology in the "basal-ganglia" brain structure [20] of vertebrates. The chosen metric space is the initial jerk of each manoeuvre *j(0)*, which represents, in the kinematic model of longitudinal driving used for the manoeuvres generation [1], the instantaneous control effort required to start the manoeuvre.

$$J_{stop} = j(0)\ of\ M_{stop}$$
$$J_{pass} = j(0)\ of M_{pass}$$

In the 1D space of *j(0)*, we define thresholds j_{th} in which the manoeuvres are considered feasible. It is interesting to mention that, according to [19], thresholds can be also associated with a probability of acting, since the probability distribution of longitudinal instantaneous jerk is evaluated experimentally.

Once the jerk set spanned by the manoeuvres set is calculated, a warning is issued if no action is available in the feasible action space $\mathcal{F}_i = [j_{th}, -j_{th}]$. Several warning level can be used associated to different thresholds, the higher the threshold, the higher priority of the warning to be issued. In the example of this paper two levels of warning are issued. In [1] levels of warnings where based on different manoeuvres, here the choice to use the jerk sets is motivated by the possibility of personalization. An example of action priming is depicted in Figure 3. In the left plot there are the primitives, the blue set is the set M_{pass}, which depends on the time slots, the green set is M_{stop}. In the right plot there is shown the jerk set composed of the union of the two subsets of the manoeuvres. Dashed lines represent respectively the borders of the area within the thresholds. In this area, if no manoeuvres are present, the respective warning is issued (grey correspond to warning level 1, and red on level 2).

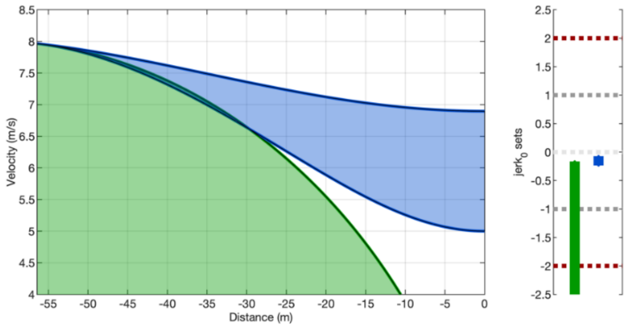

Figure 3 example of action priming

The action priming block can also generate safe manoeuvres since the construction of the time slots does not allow conflicts, as shown in the next section.

Scenario evaluation, time slots computation: In this section there are most of the novelties of this approach, which can deal with a complex intersection. In this phase mirroring of other vehicles consists in the estimation of the time the other vehicles may occupy the conflict area of the intersection. The estimation of this time is performed taking into account the following factors: all the possible feasible trajectories of the incoming vehicles, effective conflicts between trajectories and the right of way order in the intersection. The computation is performed applying the action priming algorithm seen in the previous section on the other vehicles given the initial conditions and taking all the manoeuvres.

At this step some assumptions are made: 1) A maximum velocity v_{max} at which the intersection can be crossed is selected, it does not have to be necessarily the speed limit of the road. 2) The trajectories associated to the same vehicle can be considered coincident out of the conflict area Q (see Figure 1 right) 3) The lanes can be approximated as sequences of clothoids and the path planning performed by the driver can be approximated as well as a sequence of clothoids [21][22].

For each vehicle approaching the intersection, all the feasible trajectories are instantiated, since the geometry of the intersection is fixed, the feasible trajectories does not change over time during the crossing. The reference trajectory is generated as a sequence of clothoids preserving continuity in curvature. The clothoids are generated using the library [23]. A single clothoid is a curve connection between two points with tangent vectors assigned, as shown in Figure 4. This choice is motivated by several facts: at first a feasible trajectory of a vehicle cannot have any discontinuity in curvature, since the steering wheel cannot be instantaneously rotated, and this condition corresponds to use curves with continuity in curvature. The second reason lies in the fact that clothoids are the natural optimal solution for path planning, in terms of amount of steering. Every path is composed with three clothoids: The first and the last coincide with the centreline of the lane, the second one, connects the other two (connection clothoid) from the endpoint of the first to the starting point of the last.

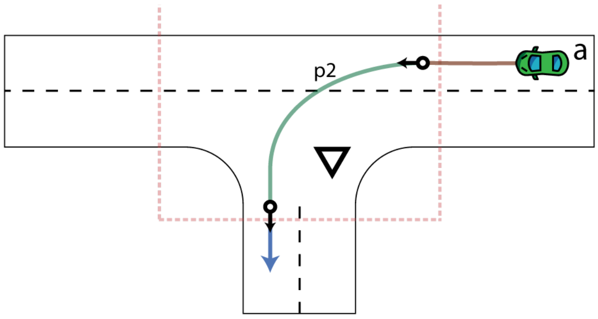

Figure 4 Example of feasible trajectory construction of one vehicle.

Once the trajectories are defined, the action priming block needs the available time slots, according to the other vehicles, the right of way order and the conflict between trajectories. The computation of the time slots is performed in several steps. This calculation requires the longitudinal states (acceleration, velocity and position w.r.t the intersection) of the vehicles involved. A scheme of the algorithm is shown at the end of the section. The first step is to instantiate one vehicle for each trajectory (from *p1* to *p6* in Figure 1), those vehicles (called virtual vehicles) are needed because the trajectory choice of each vehicle is unknow, for example the vehicle *a* can choose to follow the path *p1* or *p2* therefore the vehicle *a* is associated with the two virtual vehicles. For those virtual vehicles the points of intersection between the polygon Q (conflict area) and the trajectory p_i are defined and they are the enter points and the exit points of the conflict area. With respect to the enter point, a curvilinear abscissa *s* on the trajectory p_i is defined as a longitudinal position of the virtual vehicle, with velocity *v* and acceleration *a*. They compose the longitudinal state of the virtual vehicle, this state coincides with the one of the true vehicle associated. To resolve vehicles conflicts and inhibit overlapping time slots, the following steps are performed.

First, the possible intersections of all the trajectories are calculated. To detect the geometrical intersections, the trajectories are perturbed in order to take into account uncertainties in their computation (clothoids are approximation of the feasible trajectories). No conflict is considered between trajectories belonging to the same vehicle (in our use case the couples 1-2,3-4,5-6). After this step, the right of way order is taken into account to establish the order of passing for each conflict. In our use case, it was chosen to use the "priority to the right" rule, and the yield road signal for the vehicle *c*. To represent the conflicts, the right of way matrix is defined (R_w). The concept of right of way matrix is often use in traffic simulators (e.g. SUMO [24]). The R_w matrix is a square matrix, where the columns and the rows represent the possible paths (and then virtual vehicles). The entry $R_w(i,j)$ of the matrix represent the conflict state between the trajectory p_j and the trajectory p_i. The convention is shown below with the specific matrix of our use case.

$$R_w = \begin{pmatrix} 0 & 0 & 0 & 0 & 0 & -1 \\ 0 & 0 & +1 & +1 & 0 & -1 \\ 0 & -1 & 0 & 0 & -1 & -1 \\ 0 & -1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 \\ +1 & +1 & 1 & 0 & 0 & 0 \end{pmatrix} \quad R_w(i,j) = \begin{cases} 0 & no\ intersection \\ 1 & trajectory\ j\ has\ the\ right\ of\ way\ on\ i \\ -1 & trajectory\ i\ has\ the\ right\ of\ way\ on\ j \end{cases}$$

The matrix is antisymmetric, since for each conflict only one trajectory can have the right of way on the other. Note that the right of way matrix can be computed using a rule or provided by the infrastructure with the roads geometries. For instance, the path p_2 intersects paths p_3,p_4 and p_6. In the first two case the trajectory must yield, in the last case the trajectory has the priority. In our application the right of way order is meant to be provided by the infrastructure. The right of way matrix now it is used to inhibit the time slots of each vehicle cutting them with the ones of possible vehicles with higher priority, in order to facilitate this operation, the inhibition of the time slots must be performed in a specific order: for each trajectory the time slots with higher priority must be available for the computation. This condition can be formalized as follow: *Given* $\boldsymbol{\sigma}$ *a vector of elements indicating the order of computation for the paths* p_j*. Given* $\boldsymbol{i}$**(j) the vector of indexes of the entries with value -1 in the vector composed by the column j. The following condition must hold:*

$$\sigma_{i^*(j)} < \sigma_j \ \forall\, j, \forall\, i^*(j)$$

The condition above is equivalent to reorder the column of the matrix $\mathbf{R}_w$ in the order $\boldsymbol{\sigma}$ to obtain the resulting Matrix $R_w(i, \sigma_j)$ having -1 only in the upper triangular part of the matrix. The order $\boldsymbol{\sigma}$ is the order at which the time slots must be computed to have time slots of higher priority vehicle always available to cut those with lower priority. In our use case the order $\boldsymbol{\sigma}$ is: $\boldsymbol{\sigma} = (1,3,4,2,5,6)$.
Different vector $\boldsymbol{\sigma}$ are possible if there are no pairs of paths with no conflicts. Moreover, those vectors do not necessarily represent the order of passing of pass, the vehicles that has to yield to others can pass before respecting the time slots, and this happens if the priority vehicles are far enough. The next step is to define the maximum and minimum velocity in order to generate feasible initial time slots, this is performed using the same algorithms in [1] starting from one wide time slot (0-100s). Here the trajectories data are used to discriminate the v_{max} value: the maximum velocity is computed according to a human tolerance criteria called two-third law shown in [25] which depends on the curvature (ρ). It is chosen (according to [25]) to represent the 99.9 percentile of the human drivers. The maximum velocity is limited on a feasible one (can be higher than the legal limit of the road).

$$\begin{cases} v_{\max} = \min\left(v_{max}^{initial}, v_\rho\right) \\ v_\rho = \dfrac{3.7}{\rho^{1/3}} \end{cases}$$

At this point time slots can be initialized and then cut for all the virtual vehicles in order $\boldsymbol{\sigma}$. The first initialization takes into account only the provided bounds on velocity like the intersection was free. The inhibition consists in cut the time slots of the higher priority virtual vehicles. The time inhibited in this phase is only the set of the possible *arriving* time of the higher priority vehicles, representing all the instant they can arrive at the intersection. The time to pass the intersection must be taken into account in the inhibition (T_{cross} in Figure 5). For the ego vehicle to pass, the first time slots to inhibit is the one that the ego vehicle itself need to clear the intersection. This slot must be concatenated before any time slot of the other vehicles in order to ensure the intersection to be clear if the ego vehicle passes before others with higher priority. The time to cross of the other vehicles must be instead concatenated immediately after the last arriving time of them because the ego vehicle has to wait the other vehicles to clear the intersection to pass. The time to cross is estimated using the current velocity of the vehicle over the residual arc of the trajectory in the conflict area. This choice was made to be robust in case of unexpected sudden slowdown in the vehicle in the conflict area.

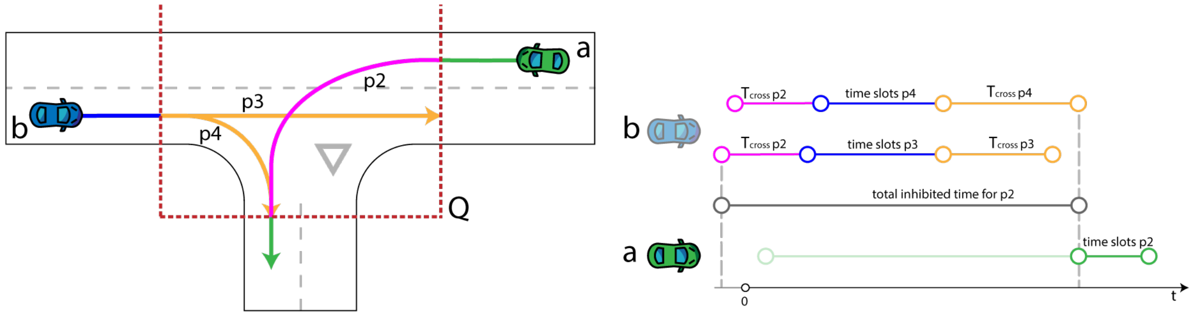

Figure 5 Example of inhibition of time slots for virtual vehicle 2

The time slots computed, are used for the computation of the manoeuvres and the warnings. The algorithm allows to compute time slots for each virtual vehicle involved and for the action priming block the time slots are the one of the ego vehicle. The algorithm is summarized here:

Algorithm 1: Intersection support for vehicle j

1: Get all vehicles states (s_j, v_j, a_j) $\forall j$

2: Instantiate virtual vehicles and trajectories and assign states

3: Calculate trajectories conflicts

4: Calculate (or get) the $\mathbf{R}_w$ matrix and get the right of way order $\boldsymbol{\sigma}$.

5: ***For*** $\forall\, \sigma_j$

- Initialize time slots $\mathcal{T}_{\sigma_j}$ using v_{min} and v_{max} .
- Calculate $T_{\sigma_j}^{cross}$
- ***For*** $\forall\, \boldsymbol{i}^*(\sigma_j)$
  - Get $\mathcal{T}_{i^*(\sigma_j)}$
  - Add $T_{\sigma_j}^{cross}$ on $\mathcal{T}_{i^*(\sigma_j)}$ *before*
  - Add $T_{i^*(\sigma_j)}^{cross}$ on $\mathcal{T}_{i^*(\sigma_j)}$ *after*
  - Cut $\mathcal{T}_{\sigma_j}$ with augmented $\mathcal{T}_{i^*(\sigma_j)}$

6: Compute manoeuvres and warning using $\mathcal{T}_{\sigma_j}$ as time slots.

Simulation results

Simulation environment

The simulation environment was developed in Matlab (2018a), to simulate the vehicle a single-track vehicle model was used in order to simulate the dynamic. The algorithm is executed periodically every 100 ms. A scheme of the simulation environment is shown in Figure 6. Two different cases are considered for the scenario in Figure 1, the difference is in the virtual driver block. 1) Full autonomous case: in this case the vehicles are driven by the codriver, using as manoeuvre selected the first available in the safe manoeuvre set. This case is aimed to validate the effectiveness of the algorithm in terms of manoeuvres generation and time slots calculation. In this case the manoeuvre selected is the slowest one of the first time slot available. 2) Warning based case: Vehicles *a* and *b* are considered autonomous (or modelled by the codriver) and vehicle c is represented by a decision-making model of the driver. The model consists of a two cases switch. *Unaware behaviour*: the driver is unaware of the other incoming vehicles, and the initial velocity remains the same until the behaviour switches to "aware". *Aware behaviour*: when warning is issued by the codriver, the reaction time of the driver is taken into account (value based on [1]) waiting 1.5 seconds before switching on the codriver safe manoeuvre (which can be a stop manoeuvre or a pass). This case is aimed to evaluate the warning anticipation, the vehicle c was chosen because is the one with most of the cases in which it has to yield to other vehicles.

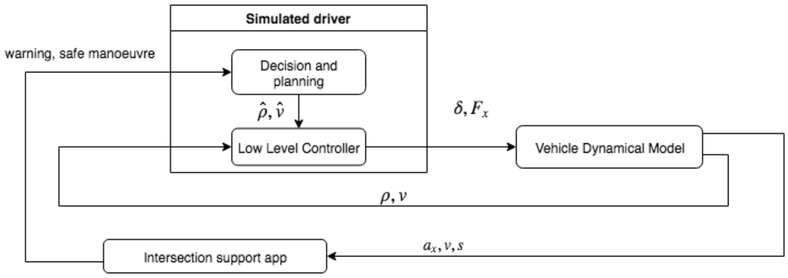

Figure 6 Simulation environment block diagram

Results

For case 1, 50 simulations were performed with random initial condition and trajectory choice. Distance and velocity from the intersection (the conflict area) were generated in the range 50-100 *m* and 8-16 *m/s,* any vehicle chooses randomly one of the two associated paths (for example vehicle 1 can choose between p1 and p2). No collisions were occurred between vehicles in the simulations, this result is shown in Figure 7. Every chart is associated with two conflict paths, (6 plots), in the y-axis the number of the experiment is indicated, and for each of them the time at which the intersection is occupied is shown. In each plot, only the case when the associated paths were selected are shown. The absence of contemporary occupancy of the conflict area is shown in Figure 7, between the crossing times is always present a safe gap, which is not explicitly imposed in the algorithm, but it raises from the fact that the time slots are considered for all the possible approaches velocities, to be robust to the effective manoeuvre choice. In the majority of the cases, the manoeuvre with right of way (the blue one) pass before the other. The opposite situation happens only with a large time gap is availabe, because the yielding vehicle considers its own time to cross.

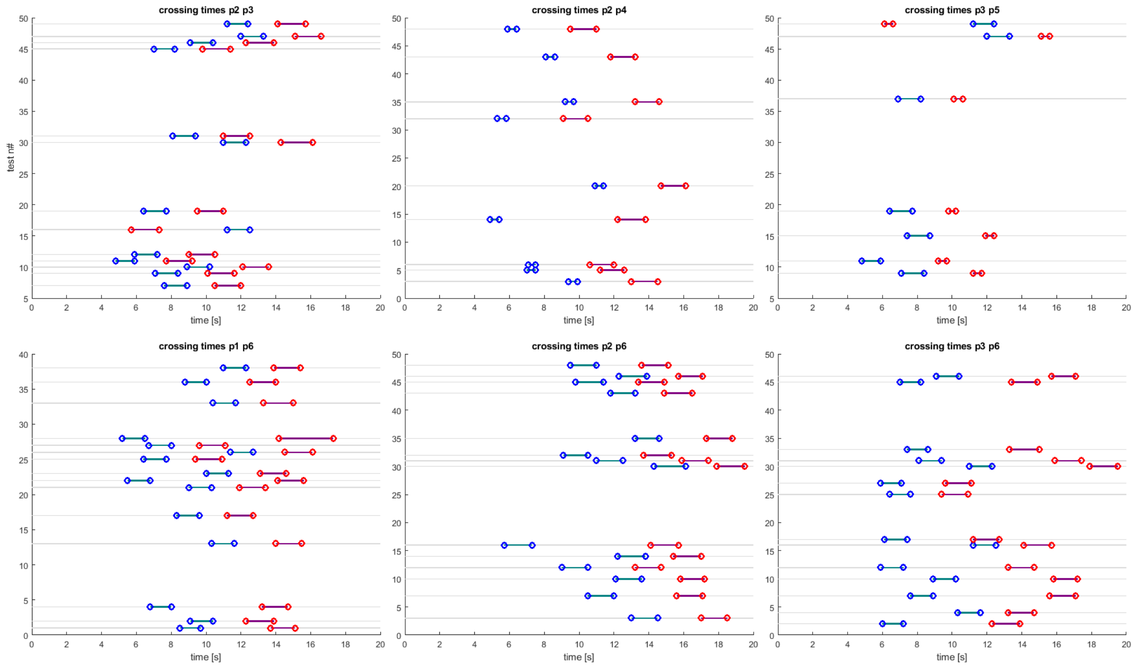

Figure 7 Crossing times for each case of conflict, in blue the manoeuvres with right of way on the others.

For case 2, the 50 simulations have the same random initial conditions, except for the distance which is generated in the range 100-200 m. In Figure 8 is shown the time to intersection (TTI) at the instant the warnings is issued. TTI is defined as the time to reach the bound of the conflict area Q keeping the same velocity: $TTI = s_f/v_0$. In Figure 8, blue triangles represent case in which the driver is the last passing the intersection, the green represents cases in which he/she passes before at least another vehicle. In most of the cases the TTI is linear with the velocity, which is the case of the theoretical warning time in [1]. Cases in which the driver can pass before, occurs when the warning is generated but there are available time slots to pass before. This can happen only if the driver has high velocity and the others are slow. There are 2 cases of TTI < 2.5 seconds, those are cases in which the intersection is rapidly cleared by the others, but the velocity is too high for the curvature of the trajectory, consequence of the implementation of the two-third law [25] for the accepted velocity.

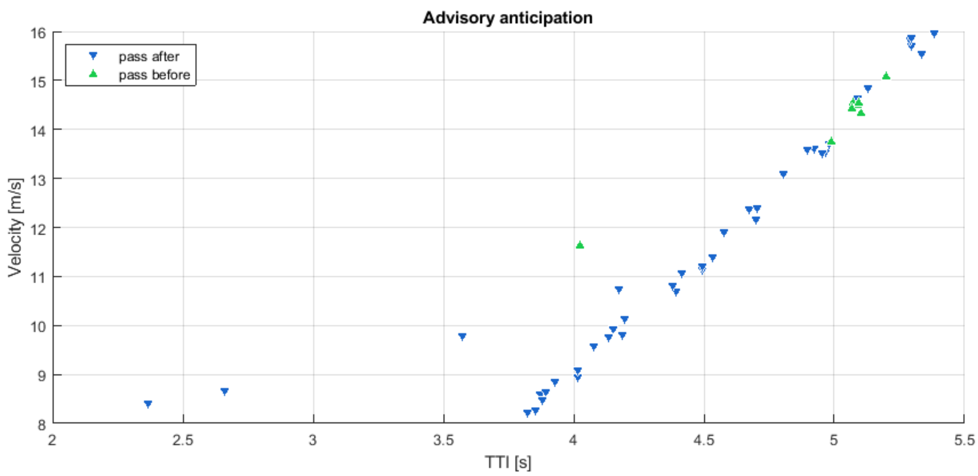

Figure 8 Time To Intersection at which warnings are issued, at j_{th} = 1 m/s^3

Conclusions

This paper presents an approach which can deal with complex intersections and multiple vehicles, introducing several novelties on the baseline work. A systematic decentralized approach to deal with conflicts using time slots, which takes into account also the traffic rules is proposed. Human-like criteria were used starting from the longitudinal motion planning to the maximum speed choice related to the curvature. The consistency of the manoeuvres generated is validated in simulation as well as the anticipatory characteristics of the warnings. For the latter experimental validation is required.

Aknowledgement

The project 2020 SAFE STRIP has received funding from the European Union's Horizon 2020 Research and Innovation Programme under grant agreement no 723211.

References

[1] M. Da Lio, A. Mazzalai, and M. Darin, "Cooperative Intersection Support System Based on Mirroring Mechanisms Enacted by Bio-Inspired Layered Control Architecture," *IEEE Trans. Intell. Transp. Syst.*, vol. 19, no. 5, pp. 1415–1429, May 2018.

[2] J. Rios-Torres and A. A. Malikopoulos, "A Survey on the Coordination of Connected and Automated Vehicles at Intersections and Merging at Highway On-Ramps," *IEEE Trans. Intell. Transp. Syst.*, vol. 18, no. 5, pp. 1066–1077, May 2017.

[3] L. Chen and C. Englund, "Cooperative Intersection Management: A Survey," *IEEE Trans. Intell. Transp. Syst.*, vol. 17, no. 2, pp. 570–586, Feb. 2016.

[4] L. Li, D. Wen, and D. Yao, "A Survey of Traffic Control With Vehicular Communications," *IEEE Trans. Intell. Transp. Syst.*, vol. 15, no. 1, pp. 425–432, Feb. 2014.

[5] I. H. Zohdy and H. A. Rakha, "Intersection Management via Vehicle Connectivity: The Intersection Cooperative Adaptive Cruise Control System Concept," *J. Intell. Transp. Syst.*, vol. 20, no. 1, pp. 17–32, Jan. 2016.

[6] M. Ahmane *et al.*, "Modeling and controlling an isolated urban intersection based on cooperative vehicles," *Transp. Res. Part C Emerg. Technol.*, vol. 28, pp. 44–62, Mar. 2013.

[7] L. Li and F.-Y. Wang, "Cooperative Driving at Blind Crossings Using Intervehicle Communication," *IEEE Trans. Veh. Technol.*, vol. 55, no. 6, pp. 1712–1724, Nov. 2006.

[8] B. Liu and A. El Kamel, "V2X-Based Decentralized Cooperative Adaptive Cruise Control in the Vicinity of Intersections," *IEEE Trans. Intell. Transp. Syst.*, vol. 17, no. 3, pp. 644–658, Mar. 2016.

[9] J.-A. Jang, K. Choi, and H. Cho, "A Fixed Sensor-Based Intersection Collision Warning System in Vulnerable Line-of-Sight and/or Traffic-Violation-Prone Environment," *IEEE Trans. Intell. Transp. Syst.*, vol. 13, no. 4, pp. 1880–1890, Dec. 2012.

[10] Y. Demiris, "Prediction of intent in robotics and multi-agent systems," *Cogn. Process.*, vol. 8, no.

3, pp. 151–158, Sep. 2007.

[11] A. Doshi and M. M. Trivedi, "Tactical driver behavior prediction and intent inference: A review," in *2011 14th International IEEE Conference on Intelligent Transportation Systems (ITSC)*, Washington, DC, USA, 2011, pp. 1892–1897.

[12] G. S. Aoude, V. R. Desaraju, L. H. Stephens, and J. P. How, "Behavior classification algorithms at intersections and validation using naturalistic data," in *2011 IEEE Intelligent Vehicles Symposium (IV)*, Baden-Baden, Germany, 2011, pp. 601–606.

[13] J. Gibson, *The theory of affordances The Ecological Approach to Visual Perception (pp. 127-143)*. Boston: Houghton Miffin, 1979.

[14] G. Hesslow, "The current status of the simulation theory of cognition," *Brain Res.*, vol. 1428, pp. 71–79, Jan. 2012.

[15] S. Thill, D. Caligiore, A. M. Borghi, T. Ziemke, and G. Baldassarre, "Theories and computational models of affordance and mirror systems: An integrative review," *Neurosci. Biobehav. Rev.*, vol. 37, no. 3, pp. 491–521, Mar. 2013.

[16] J. Decety and J. Grèzes, "The power of simulation: Imagining one's own and other's behavior," *Brain Res.*, vol. 1079, no. 1, pp. 4–14, Mar. 2006.

[17] S. Hurley, "The shared circuits model (SCM): How control, mirroring, and simulation can enable imitation, deliberation, and mindreading," *Behav. Brain Sci.*, vol. 31, no. 01, Feb. 2008.

[18] M. Da Lio *et al.*, "Artificial Co-Drivers as a Universal Enabling Technology for Future Intelligent Vehicles and Transportation Systems," *IEEE Trans. Intell. Transp. Syst.*, vol. 16, no. 1, pp. 244–263, Feb. 2015.

[19] M. Da Lio, A. Mazzalai, K. Gurney, and A. Saroldi, "Biologically Guided Driver Modeling: the Stop Behavior of Human Car Drivers," *IEEE Trans. Intell. Transp. Syst.*, pp. 1–16, 2017.

[20] R. Bogacz and K. Gurney, "The Basal Ganglia and Cortex Implement Optimal Decision Making Between Alternative Actions," *Neural Comput.*, vol. 19, no. 2, pp. 442–477, 2007.

[21] E. Bertolazzi and M. Frego, "On the G 2 Hermite Interpolation Problem with clothoids," *J. Comput. Appl. Math.*, vol. 341, pp. 99–116, Oct. 2018.

[22] E. Bertolazzi and M. Frego, "Interpolating clothoid splines with curvature continuity," *Math. Methods Appl. Sci.*, vol. 41, no. 4, pp. 1723–1737, 2018.

[23] E. Bertolazzi and M. Frego, "G1 and G2 fitting with clothoids, circle and biarc." MATLAB Central File Exchange, Jun-2018.

[24] J. Erdmann and D. Krajzewicz, "SUMO's Road Intersection Model," in *Simulation of Urban Mobility*, vol. 8594, M. Behrisch, D. Krajzewicz, and M. Weber, Eds. Berlin, Heidelberg: Springer Berlin Heidelberg, 2014, pp. 3–17.

[25] P. Bosetti, M. Da Lio, and A. Saroldi, "On Curve Negotiation: From Driver Support to Automation," *IEEE Trans. Intell. Transp. Syst.*, vol. 16, no. 4, pp. 2082–2093, Aug. 2015.

[26] H2020 Safestrip project http://safestrip.eu