Collision-Free Intersection Crossing of Mobile Robots Using Decentralized Navigation Functions on Predefined Paths

Benjamin Fankhauser, Laleh Makarem and Denis Gillet

Abstract— This paper deals with the coordination of a group of mobile robots at an intersection. It focusses on decentralized navigation functions (DNFs) to achieve efficient traffic control. The main challenge is to define virtual potentials, which are used by decentralized navigation functions, such that traffic is both fluent and safe, while taking into account real-world limitations like acceleration, braking and speed limits. Our method consists in defining the navigation function with respect to the desired acceleration profile and is accompanied by a set of visibility conditions that increase the capacity of the intersection in terms of vehicle throughput. Priority conditions have been used to both avoid blockades of robots and to save energy by assigning higher priorities to robots with higher inertias.

I. Introduction

Compared with a single robot, multiple mobile robots can introduce additional advantages in terms of flexibility, robustness, efficiency and comfort. Therefore there has been a considerable interest in the field of mobile robot interaction in recent years. Coordination of a group of mobile robots has been mentioned several times in literature due to its increasingly important role in military, commerce and science. Impressive works in the field of coordination, presented at [1], [2], successfully exploited knowledge from mobile robotics. Accordingly, it could be expected that in the near future autonomous or semi-autonomous driver assistance systems will be available to handle traffic at intersections [3]. This paper addresses the problem of intersection crossing by mobile robots. This problem can be generalized to that of the autonomous crossing of vehicles taking into account complexities of vehicles in dynamics and tasks. Different control approaches for autonomous navigation have been proposed in the literature [4]. One approach is to describe the robots using kinematic models, with the objective of reaching a destination while avoiding collisions with fixed and moving obstacles. In this approach, autonomous navigation is seen as a multi-objective problem for which various solutions have been suggested; such as stochastic optimization [5], cooperative methods of control [6], [7], and decentralized control [8]. Among all these methods, decentralized control has so far received more attention as a method that does not need long-range communication and shows more robustness when it comes to various communication failures in the system [9]. The use of navigation functions in the decentralized scheme seems promising, as its implementation is feasible in real-time and as navigation functions also show good flexibility with regard to adding vehicles and a changing environment [10].

In this paper, the focus is on developing a flexible computationally efficient decentralized algorithm for mobile robots at intersections while considering the different inertias of the robots and their priority for passing, as well as their ability to brake and accelerate at the intersections. As such, the robots need the ability to communicate and transfer data about their positions and their inertia. Our proposed navigation function is initially based on the distance of every robot from its goal destination, while the navigation is restricted to a certain path. In a second stage, it is based on the distance between the different robots, their respective speeds and the directions they are heading in. A visibility zone is defined around each vehicle in order to emulate a real detection and communication range. We paid special attention to intersections because they correspond to traffic conditions potentially having a high impact on energy consumption and motion smoothness.

In the second section of this paper, a dynamical model of the vehicles is introduced. It is simple enough to enable the handling of complex traffic situations and complex enough to enable energy optimization. In the third section, a decentralized navigation function that enables taking dynamical constraints into account is proposed. The evaluation of the proposed approach is presented in section 4. It is compared with a rule-based crossing method and traditional crossing of manned vehicles at an intersection with traffic lights. Finally, concluding remarks and outlook are given in the last section.

II. Problem Formulation

The considered multi-robot system consists of N mobile robots. The goal of each robot is to cross an intersection without having any collision with other robots (Figure 1).

The position of robot i is known as $q_i = (x_i, y_i)$ in a global frame attached to the intersection. The path of the robot is predefined and could be described by path parameter s_i. Hence, the position of a robot in the global frame could be calculated from its location along the path using the parametric function $q_i = f_k(s_i)$ corresponding to the path k the robot has chosen. The motion of each robot along its path is modeled using second order dynamics:

$$\ddot{s}_i = a_i \tag{1}$$

Note that the dynamics proposed for every robot is quite realistic as the assumption of predefined routes is valid. Additionally, we introduce real world limitations like an acceleration limit a_{max}, as well as a braking limit b_{max}.
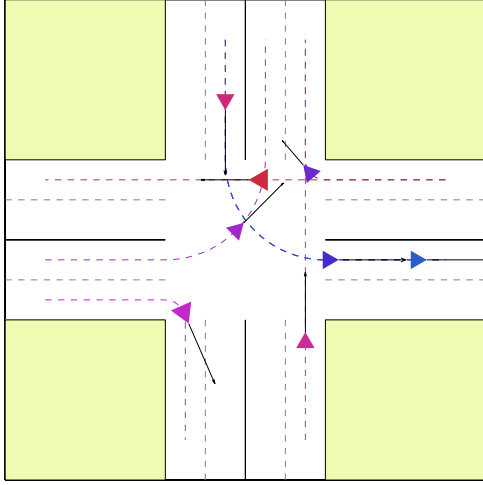
B. Fankhauser, L. Makaraem and D. Gillet are with Real Time Coordination and Distributed Interaction Systems (React) Group, Ecole Polytechnique Federale de Lausanne (EPFL), 1015Lausanne, Switzerland `laleh.makarem@epfl.ch`, `Denis.Gillet@epfl.ch`

Fig. 1. The goal of each robot is to cross an intersection without having any collision with other robots.

Fig. 2. Possible navigation function in a four-way double lane scenario as perceived by the green mobile robot on a left turning path. The negated gradient draws the robot towards the goal and repels it from the blue robot.

The speed limit is given by the function $v_{max} = v_{lim}(s_i)$ such that the centripetal acceleration in the bends remains below a certain value. The speed $\dot{s}_i$ is bounded to the interval $[0, v_{max}]$.

The problem is now to find a navigation function that guarantees safety and high capacity while facing the mentioned real-world limitations related to acceleration, speed and heading.

III. DECENTRALIZED NAVIGATION FUNCTION

A navigation function is a smooth mapping from a working manifold of mobile robots to a scalar which should be analytic in the workspace of every robot and its gradient would be attractive to its destination and repulsive from other robots. So, an appropriate navigation function could be combined with a proper control law in order to obtain a trajectory for every mobile robot leading to the destination and avoiding collisions. Decentralization in the navigation function has been introduced in [11]. The key idea in a decentralized navigation function is taking into account the limited sensing and the individual control of each robot. This decentralized process significantly improves the applicability and scalability of the algorithm in larger scenarios as it can take into account the finite range of real sensors.

Let us define a navigation function as follow:

$$\phi_i = a_d(pl - s_i) + \sum_{i \neq j} \beta(q_i, q_j, ...) \tag{2}$$

where pl is the length of the path between the initial and the goal position and a_d the desired acceleration.

We can see that the first term leads to a linear attraction of the mobile robot towards the goal with an acceleration of a_d. So, in the absence of other mobile robots, each robot will accelerate toward its goal, until it reaches its acceleration limit. Thus, the problem is reduced to finding a convenient expression for the β-function in the second term, which will repel the robot from nearby robots.
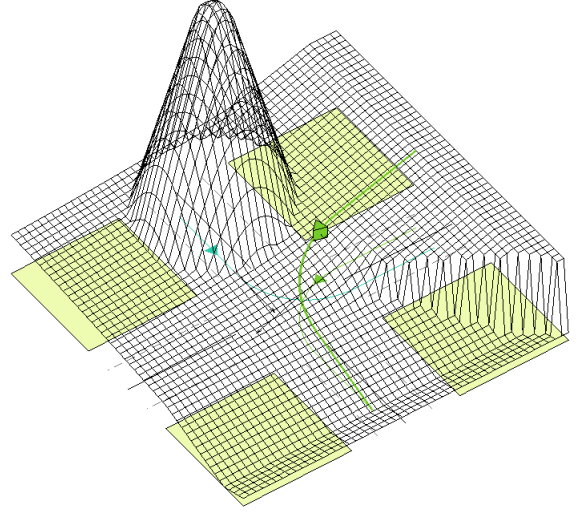
The main objective in the coordination of mobile robots is to guarantee that no collision between robots will occur. In addition, one could propose other goals such as energy efficiency and lower communication costs. In order to be safe and efficient, the robot should not brake too early, but quickly enough to stop just in front of its nearby robot. This defines a certain security zone with radius x_s around the robot. As soon as the nearby robot enters this zone, the robot will have to brake. The manner in which the robot brakes depends on the form of the β-function. This means that one can define the form of the β-function so that the robot brakes at the right time according to its velocity and the distance from the nearby robot.

A. Decentralized control of each robot

The system is controlled by Decentralized Navigation Functions, specific to every robot. Such a navigation function ϕ_i can be seen as a potential field whose negated gradient is attractive towards the goal and repulsive from other robots (figure 2). According to the navigation function presented in 2 and the dynamics of robots defined in 1, this gradient is then conveniently used as control input:

$$\vec{u}_i = -\nabla \phi_i \tag{3}$$

As every robot is supposed to move along its pre-defined path, the proposed force should be projected along the path. Assuming that the robot has no deviation from path the force could be projected along the heading of the robot.

$$a_i = \vec{h}_i \cdot \vec{u}_i \tag{4}$$

This corresponds to the projection of the control input $\vec{u}_i$ onto the unit vector $\vec{h}_i$ indicating the heading of the robot.
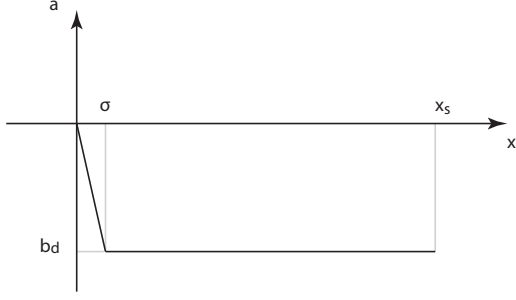
Fig. 3. Desired Acceleration Profile: The braking intensity grows linearly over a short distance σ, after which it remains constant. The distance x_s designates the distance that is needed for safe deceleration when encountering another robot.

B. Collision avoidance

First of all, let us consider the problem of a mobile robot that encounters another robot at rest in one dimension. Let us further assume an acceleration profile as shown in figure 3. Basically, the profile corresponds to a constant deceleration b_d, to which we have added a linear transition over a relatively small distance σ to avoid abrupt changes in acceleration. In order to avoid a collision, it becomes clear that all the kinetic energy of the mobile robot must be dissipated by braking. Thus we can write the following equation:

$$\frac{1}{2}mv_i^2 = mb_d\left(x_s - \sigma\right) \tag{5}$$

Note that we do not consider the amount of energy dissipated over the distance σ. This can be seen as a sort of additional safety measure and simplifies our calculations. The security distance will thus be given by:

$$x_s = \frac{v_i^2}{2b_d} + \sigma \tag{6}$$

As shown in equation 3, the acceleration will be the spatial derivative of the navigation function in equation 2, assuming that we still consider one dimension and that the projection that is required by equation 4 therefore has no effect. This, in turn, means that the navigation function must be the spatial integration of the acceleration profile shown in figure 3 and which allows us to calculate the β-function:

$$\beta = \begin{cases} (a_d + b_d)\left(x_s - d - \frac{\sigma}{2}\right) & if\ d < x_s - \sigma \\ (a_d + b_d)\frac{(x_s - d)^2}{2\sigma} & if\ x_s - \sigma > d > x_s \\ 0 & otherwise \end{cases} \tag{7}$$

where $d = |x_i - x_j|$ is the distance between the mobile robots and x_s is given by equation 6.

Let us suppose now that the nearby robot j is not at rest, but rather that it's coming towards the robot with speed v_j. This changes the kinetic energy of the system in equation 5 in the sense that we have to add the kinetic energy of the nearby robot. As expected, the security distance becomes
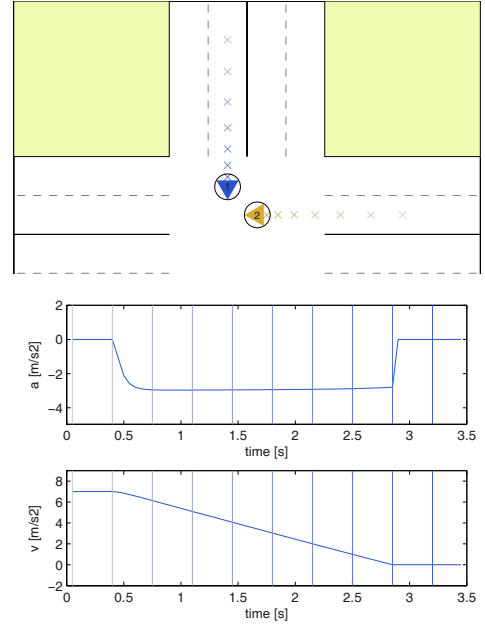

Fig. 4. Two mobile robots braking at a desired deceleration of $b_d = 3$ m/s^2 in order to avoid collision.

bigger i.e. the robot starts to brake earlier:

$$x_s = \frac{v_i^2}{2b_d} + \frac{v_j^2}{2b_d} + \sigma \tag{8}$$

The extension to two dimensions can be done by projecting the motion of two mobile robots onto the vector $\vec{d}$ that separates them from each other. This projection allows us to compare the movement of the robots in one dimension and we can thus apply the afore mentioned method. There is one restriction: the heading vector $\vec{h}_i$ in equation 4 is not necessarily in line with the control vector $\vec{u}_i$. This means that the acceleration a_i might differ from the norm of the control vector $\vec{u}_i$ by an amount of $\cos(\delta_i)$, which is the cosine of the angle between $\vec{h}_i$ and $\vec{u}_i$. To solve this problem we divide the amount of the β-function by $\cos(\delta_i)$. The result is shown in figure 4.

Note that we chose to use an acceleration profile that depends on distance rather than time. This choice of acceleration profile considerably simplifies the integration to obtain the β-function without loss of generality.

C. Visibility conditions

As we could show by simulation in MATLAB, the method presented in the previous paragraph allows for safe navigation with DNFs in general. Because we adapted predefined paths, however, some of the safety measures are superfluous and slow the traffic unnecessarily (e.g. if two mobile robots are driving next to each other on parallel lines). We therefore introduced three conditions according to which a nearby robot needs to be visible or not.

First, we consider that the field of vision of a mobile robot is 1.4 radians to the left and 1.4 radians to the right, any
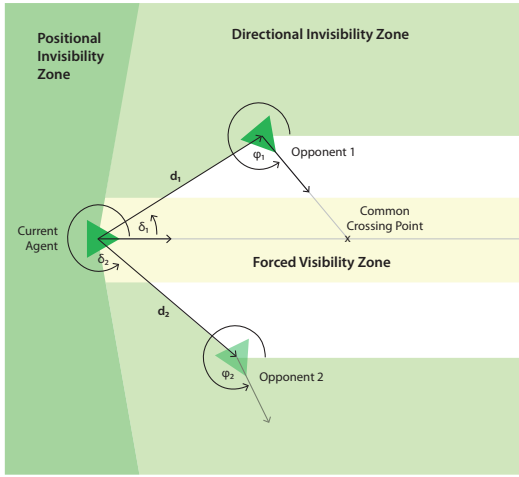
Fig. 5. The nearby robot is only visible if there is a common crossing point coming up or if he's situated in the Forced Visibility Zone.

nearby robot outside this zone is invisible:

$$cnd_1 = 0 \le \delta \le 1.4 \, and \, \pi + \delta < \varphi < 2\pi \qquad (9)$$

Second, as shown in figure 5, there is no danger of collision if the heading vector $\vec{h_j}$ of the nearby robot is inside the light green zones. Which means the nearby robot will keep out of the way of the robot. So, it should be considered as no risk or invisible. (e.g. Nearby robot 2):

$$cnd_2 = 2\pi - 1.4 \le \delta \le 2\pi \, and \, 0 < \varphi < \delta - \pi \qquad (10)$$

Third, if the nearby robot is inside the Forced Visibility Zone, it should be visible at any rate. This is due to the fact that the robots occupy a disc of radius r_i and r_j respectively:

$$cnd_3 = |d \cdot \sin \delta| \le r_i + r_j \, and \, (1.4 < \delta < 2\pi - 1.4) \quad (11)$$

If either of these conditions is satisfied, the nearby robot should be visible.

Note that these conditions were defined for robots on straight paths. Due to the nature of our predefined paths, however, the conditions remain correct for curved paths. The only change was that we replaced $|d \cdot \sin \delta|$ in equation 11 with the real distance of the nearby robot to the (curved) path of the mobile robot.

D. Priority Assignment

So far, all robots have been treated equally. However, there are at least two good reasons to give priority to certain individuals rather than others. The most obvious is to give priority to mobile robots with higher inertia, thus saving energy. This can be implemented by multiplying the nearby robot's security distance by a certain factor, 1.5 in our case. Note that this manner of giving priority can be extended to any kind of vehicles such as ambulances or police cars in the intersection scenario. Indeed this simple implementation of priority rights is one of the important characteristics of decentralized navigation functions.

At the same time priority can also help avoid blockades of two crossing mobile robots. Relying on the previously
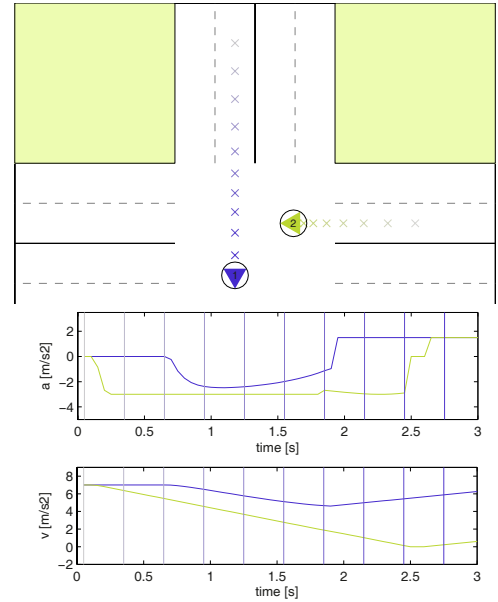

Fig. 6. Mobile robot 2 brakes earlier and lets mobile robot 1 pass, thus avoiding a blockade.

presented method, all mobile robots will avoid collision by braking. But considering the scenario in figure 4, we would have preferred one robot to brake earlier and give priority to the other, thus avoiding the blockade. The decision regarding which mobile robot is going to brake and which is let pass intuitively depends on the distance to the crossing point. The robot that is closer to the crossing point gets priority.

Instead of using the true distances as an indication, which is laborious to establish with curved paths, we use the angle δ_i between heading $\vec{h_i}$ and common distance vector $\vec{d}$. The bigger the angle, the closer the robot is to the common crossing point. As shown in figure 6 robot 2 brakes earlier and gives priority to robot 1.

In fact, in the case of an interaction of more than two vehicles, it can still happen that two robots block each others' passage. This is solved by making a nearby robot invisible to the robot as soon as the latter is in the former's path. The robot will thus accelerate, while the nearby robot will stay blocked.

IV. SIMULATION AND RESULTS

In order to measure the performance of the proposed method, we carried out a comparative evaluation by simulation with two other methods of controlling traffic in a crossing: traffic lights and a rule-based algorithm.

The traffic lights are initialized in a way that in the first scenario (only straight-heading paths) the green light is given to two opposing lanes during a time $t_{min} = 8.6 \, s$ after which the light is made to change if there are more vehicles waiting in the opposing lanes than in the current one. The maximum waiting time is defined as $t_{max} = 2t_{min}$. In the second scenario (multi-directional paths), the green light is given to two neighboring, occupied lanes during t_{min}. If one of the lanes contains no waiting vehicle, an occupied alternative

Fig. 8. Energy Consumption With Respect to Priority Conditions. We can see that if priority is given to the heavier lorries which represent 25 % of the vehicles, the total energy of all vehicles crossing the intersection is lower.
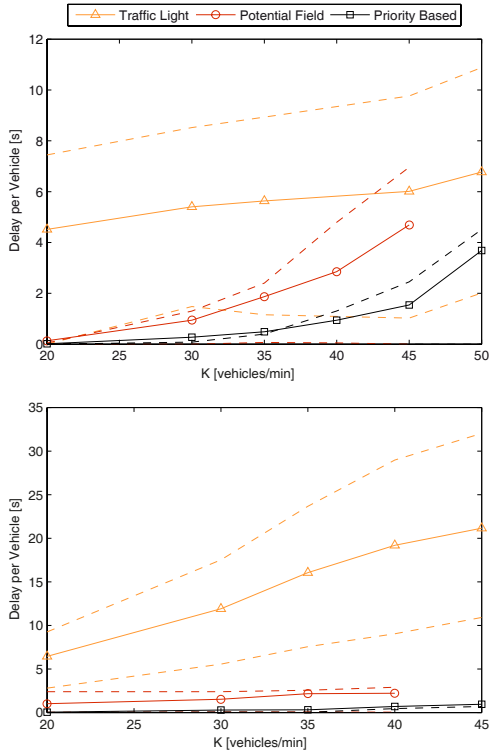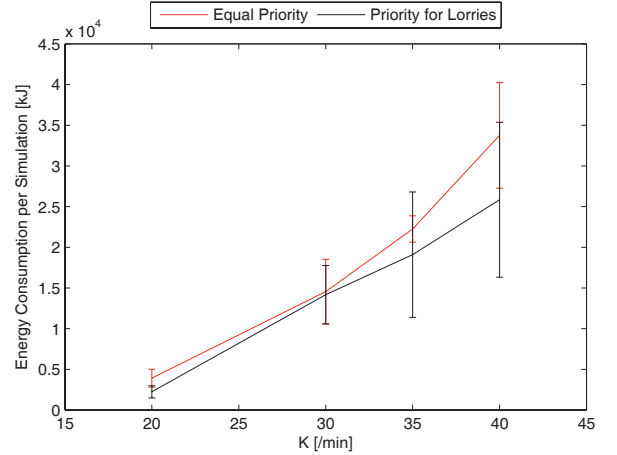
Fig. 7. Average Delay per Vehicle on Straight-Heading (above) and Multi-Directional (below) Paths. The dashed lines show the 25% and 75% quantiles as the error is not parametric. In the second scenario the vehicles leave the crossing up to 17 seconds earlier controlled by DNFs as compared to a traffic light control, while 25% of them are not slowed down at all.

lane is chosen instead. The maximum waiting time can be calculated as $t_{max} = 3\,(t_{min} + t_{ch})$, where $t_{ch} = 3.2\,s$ is the time that the lights need to change from red to green.

The rule-based algorithm gives priority to the mobile robot closer to the common crossing point, without slowing it down as in the case of robot 1 in figure 6, whereas the nearby robot is made to brake right in front of the crossing point. In one time step when all the robots are in the vicinity of the intersection the decision is made by all the present robots. The actions of robots will not change before the first one successfully passes the intersection. This method lacks the robustness of the decentralized navigation function, but allows for an almost optimal traffic flow.

The following results are all based on a series of six simulations of five minutes. Thus, each value is determined by a total of 30 minutes of simulation. Note that we interrupt the simulation if more than 80% of the vehicles are at rest, assuming that in this case there is a blockade of more than two vehicles. The vehicles, of which 75% are cars (r=1 m, m=1.2 Mg) and 25% lorries (r=1.2, m=20 Mg), are initialized at regular intervals according to the number K of vehicles per minute. Every vehicle is assigned randomly to one of the different paths. All the paths have equal probability. In a first step we give priority only according to the distance to the common crossing point and make no distinction between cars and lorries.

We measured the performance in terms of both delay and energy. By delay we understand the difference between the time the vehicle actually needs to cross and the minimal time in which this would be possible if the vehicle was alone in the crossroad. A comparison in terms of average delay per vehicle (figure 7) shows that the DNFs are performing very well with respect to traffic lights and for lower numbers of vehicles. Indeed 25 % of the vehicles are not slowed down at all. In the first scenario the difference is at 3.7 seconds per vehicle at K=35, in the second it is even more striking with 17 seconds per vehicle at K=40. This difference between the two scenarios is due to the fact that traffic lights perform less efficiently with multi-directional paths, whereas the performance of DNFs shows no dependency on the nature of the paths. Considering the crossroad as a whole we can say that by using DNFs in the second scenario,12 minutes of the drivers time could be saved in every crossing of intersection. A comparison with the rule-based algorithm shows that the performances of the DNFs are closer to the optimal performance of rule-based algorithm than to the traffic lights.

The energy consumption was measured by calculating the work of acceleration or braking for every vehicle at every time step. The results are similar to those of the delay, which was to be expected. Clearly, the vehicles are delayed, because they brake and if they brake, they dissipate energy. Table I shows the percentage of energy that could be saved by using DNFs rather than traffic lights. We see that for multi-directional paths we could save between 30 - 50 % of the energy.

In a second step we looked at the scenario with straight paths, but introduced a priority rule for the heavier lorries. The effect of this easily implementable priority rule is that the energy consumed in the crossroad is lower (figure 8) and the whole method becomes more energy efficient. More

K [/min]	Saved Energy	
	Scenario 1	Scenario 2
20	75 %	49 %
30	51 %	39 %
35	37 %	34 %

TABLE I

PERCENTAGE OF ENERGY THAT CAN BE SAVED BY USING DNFS RATHER THAN TRAFFIC LIGHTS. FOR MULTI-DIRECTIONAL PATHS (SCENARIO 2), WE COULD SAVE BETWEEN 30 - 50 % OF THE ENERGY.

extended simulations with network of intersections would allow to determine a more significant difference.

Note that at $K = 45$ (1st scenario) and $K = 40$ (2nd scenario) we start to have blockades of more than two robots, for which a solution remains to be found.

V. DISCUSSIONS

The proposed DNF method for controlling multiple mobile robots at an intersection has been tested in simulation and compared to the classical traffic light control and a rule-based crossing method. Assuming only straight paths, we have found that compared to traffic lights, the DNF method is more efficient in terms of time and energy for a lower flow of vehicles. We have also found that between 35-75 % of the consumed energy could be saved with respect to traffic lights and that the performance of DNFs is quite close to that of the rule-based control method. It's worth mentioning that the rule-based method may introduce a higher energy efficiency but is not practically possible in real world. We also showed, using DNFs, that 25 % of all vehicles did not have to be slowed down at all.

Allowing for multi-directional paths in a second scenario, we found that the DNF method performed even better in terms of delay. Thus we showed that for 40 vehicles per minute, a vehicle would quit the crossroad on average 17 seconds before one that had been controlled by traffic lights. This meant a saving of 12 minutes on the whole crossroad. Because the traffic lights made the vehicles brake only once, we realized that the difference in performance is not so outstanding with regards to energy, but still about 30-50% of the energy could be saved by using DNFs rather than traffic lights. In fact, this figure could well be increased if one considers that vehicles that are waiting in the queue of a traffic light are also consuming energy.

To conclude, we might say that in addition to the general advantages of robustness and easy computation much speaks in favor of using DNFs as a method of controlling the traffic in crossings, both in terms of delay and energy-efficiency. To be able to use them, however, a solution to overcome the blockades caused by more than two crossing vehicles must be found.

VI. CONCLUSION AND FUTURE WORK

With the goal of traffic control at intersections, we have introduced a method to guarantee safety using decentralized navigation functions for mobile robots on predefined paths.

This work consists of modeling the navigation function with respect to the desired acceleration profile. The introduced navigation function is accompanied by a set of visibility conditions that increases the capacity of the intersection in terms of vehicle throughput. Priority conditions have been used to both avoid blockades of two mobile robots and to save energy by assigning higher priority to vehicles with greater inertia (e.g. lorries).

Our future research will include the analytical study of the convergence of the proposed coordination approach. More realistic features will be added, like more complex dynamics of the vehicles. We will also study the behavior of the vehicles under communication constraints and lack of energy as could happen when using electrical vehicles. Different profiles of acceleration and deceleration will be taken into account in order to have a comfortable driving experience. This will allow us to adapt our work to autonomous vehicles and semi-autonomous driver assistance systems.

REFERENCES

[1] "DARPA Urban challenge", [Online]. Available: http://www.darpa.mil/grandchallenge

[2] M. Montemerlo, J. Becker, S. Bhat, H. Dahlkamp, D. Dolgov, S. Ettinger, D. Haehnel, T. Hilden, G. Hoffmann, B. Huhnke, D. Johnston, S. Klumpp, D. Langer, A. Levandowski, J. Levinson, J. Marcil, D. Orenstein, J. Paefgen, I. Penny, A. Petrovskaya, M. Pflueger, G. Stanek, D. Stavens, A. Vogt, and S. Thrun. "Junior: The Stanford Entry in the Urban Challenge" *Journal of Field Robotics*, Volume 25, Issue 9, pp. 569, September 2008.

[3] S.Thrun, "Toward robotic cars", *Communications of the ACM* Volume 53, Issue 4, 1 April 2010, pp. 99.

[4] R.M. Murray, "Recent Research in Cooperative Control of Multivehicle Systems", *Journal of Dynamic Systems, Measurement, and Control*, Vol. 129, No. 5. (2007), pp. 571.

[5] J. S. Baras, X. Tan and P. Hovareshti, "Decentralized control of autonomous vehicles" *in Proceeding of IEEE Conference on Decision and Control*, Maui, Hawaii, 2003, pp. 1532.

[6] H. Roozbehani, S. Rudaz, and D. Gillet, "A Hamilton-Jacobi formulation for cooperative control of multi-agent systems", *in IEEE International Conference on Systems, Man and Cybernetics*, 2009, pp. 4823.

[7] Y. Bouteraa, A. Mansour, J. Ghommam, G. Poisson"Cooperative control and synchronization with time delays of multi-robot systems ", *Journal of Engineering and Computer Innovations*, Vol. 2(2), pp. 28-39, February 2011.

[8] H. Roozbehani, S. Rudaz, and D. Gillet, "On decentralized navigation schemes for coordination of multi-agent dynamical systems", *in IEEE International Conference on Systems, Man and Cybernetics*, 2009, pp. 4807.

[9] M. C. De Gennaro and A. Jadbabaie, "Formation Control for a Cooperative Multi-Agent System using Decentralized Navigation Function", Proceeding of American Control Conference, Minneapolis, USA 2006.

[10] D. V. Dimarogonas, M. M. Zavlanos, S. G. Loizou, and K. J. Kyriakopoulos, "Decentralized motion control of multiple holonomic agents under input constraints", *in Proceeding of 42nd IEEE Conference on Decision and Control*, 2003, pp. 3390.

[11] D. V. Dimarogonas and K. J. Kyriakopoulos, "Decentralized Navigation Functions for Multiple Robotic Agents with Limited Sensing Capabilities", *Journal of Intelligent and Robotic Systems*, Vol. 48, No. 3, pp. 411-433, 2007