2014 IEEE 17th International Conference on
Intelligent Transportation Systems (ITSC)
October 8-11, 2014. Qingdao, China

Autonomous Lane-Change Controller via Mixed Logical Dynamical

Yaoqiong Du[1], Yizhou Wang[1] and Ching-Yao Chan[2]

***Abstract*— This paper focuses on the design of a model based methodology to optimize lane change maneuvers based on the predictions of neighboring agents. In particular, the dynamics of vehicles are modeled as double integrators, and the lane change actions are indicated by boolean variables while neglecting the lane change dynamics. The objective of the optimal control is to minimize the travel time, maintain saftey distances between the target vehicle and neighboring ones, satisfy the operational constraints of the target vehicle, and comply with the speed limit.**

The control problem is formulated as a mixed logic dynamic system, and solved by Cplex, a commercial mixed integer optimization solver. Finally, the effectiveness of the proposed methodology is demonstrated by two simulated scenarios in this paper.

I. INTRODUCTION

Autonomous vehicles are expected to significantly improve safety, efficiency, energy consumption, comfort and mobility. Ideally, such vehicles should be able to operate without requiring major road infrastructure changes, and be able to cope with the complexity of urban traffic, and share roads with human-driven vehicles and pedestrians.

One challenging task of autonmous vehicles is to cope with urban driving environment, and to make safe and appropriate driving decisions in various traffic situations. The solution to this challenge involves a high-level vehicle decision-making in control strageteties. Furthermone, the vehicle control problem can be considered in terms of trajectory generation with controllers designed for both longitudinal and lateral movements. In this paper, our discussion will be focused on the strategic decision making when lane change and overtake maneuvers are desirable and feasible, assuming that once a lane change decision has been made, a lower level controller will be able to track a pre-computed reference.

Methods for strategic decision-making in fully or highly automated driving systems designed for lane change and overtake maneuvers, can roughly be divided into rule-based [1], [2], utility-based [3], [4], and other approaches. The lane changing maneuver is carried out by planning the reference trajectory according to the vehicle states and road information, and then the control laws are designed using onboard sensors to track this virtual trajectory [5]. Hatipoglu*et. al.*. [6] reported the design of an automated lane changing controller with a two-layer hierarchical architecture. In this way, the open loop lane change problem has been converted into an equivalent virtual reference trajectory tracking problem. In the study [7], the algorithm of MPC is employed for designing an intelligent vehicle lane change controller by using nonlinear vehicle lateral movement mode and simulation software CarSim. Julia Nilsson*et. al.* [8] presents an algorithm for strategic decision making regarding By considering the task of driving on two-lane, one-way roads. Masakazu Mukai *et. al.* [9] considers an optimal lane change path generation for intelligent automobiles using hybrid system representation. The more advanced approaches also include probabilistic methods to handle uncertainties [10], [11]. And the system in study [12] uses fuzzy controllers that mimic human behavior and reactions during overtaking maneuvers. A high-precision Global Positioning System and a wireless network environment base the system on the information that is supplied.

Most of those lane changing maneuvers researched above focus on two-lane configurations only. Rule-based systems have the advantage of traceability and ease of implementation for specified scenarios but can require a substantial effort in order to be extended. On the other hand, utility-based approaches have the advantage of allowing combined weighting of multiple criteria and can thus more easily be extended to complex scenario [8]. However, a large amount of different weighting parameters can result in time-consuming parameter tuning and tractability difficulties.This paper aims at the study of the autonomous vehicle lane changing decision making on multiple lanes, and organizes as follows. Section II presents an optimization-based lane-changing decision making. Simulation models built upon the proposed approach are performed and their results are presented in Section III. Conclusions are given in Section IV.

II. OPTIMIZATION APPROACH

In this section, the lane change decision making problem is formulated as an optimization problem. Specifically, the vehicle dynamics, and safety constraints are transformed into be a mixed logical dynamical (MLD) system [13]. The optimal controller of such a MLD system is then synthesized by a mixed-integer programming (MIP) approach.

The problem of deriving decisions regarding appropriate driving maneuvers is hereby considered in the following

[1]The authors are with Department of Mechanical Engineering, University of California Berkeley dyq_cal@berkeley.edu ,yzhwang@berkeley.edu
[2]The authors are with California PATH, University of California Berkeley cychan@berkeley.edu

three aspects: 1) selection of most appropriate lane; 2) selection of velocity profile; 3) minimization of travel time under safety condition.

A. System modeling

In order to create a simplified model to represent the vehicle dynamics, a double integrator is used to govern the longitudinal motion. $x_0(t)$ and $v_0(t)$ are used to denote the longitudinal position and velocity of the controlled vehicle, respectively. The lateral motion is simply represented by a discrete state $q(t) = \{1, \dots, n\}$, indicating which lane the vehicle is occupying, where n is the total number of lanes. The two control signals under consideration are (i) the longitudinal acceleration u(t) (continuous) and (ii) the lane changing decision $\sigma(t) \in \{-1, 0, 1\}$ (discrete). $\sigma(t) = 0$ implies that the controlled vehicle stays in the current lane, and $\sigma(t) = 1$ (or -1) implies the vehicle changes to the right (or left) lane. This vehicle model can be represented by the following diagram depicting a hybrid automaton system(Figure 1).

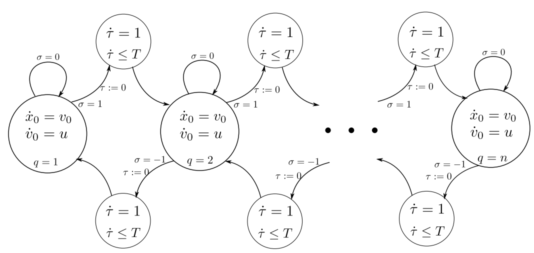

Fig. 1. The vehicle dynamics is modeled by a hybrid automaton. T is the required time for a lane changing maneuver.

For the modeling of system dynamics, there are following assumptions:

1) For the purpose of designing a decision algorithm regarding preferred lane and desired acceleration/deceleration, it is sufficient to represent the vehicle as a point-mass.This simplification is adequate since the focus of the algorithm is on decision making regarding appropriate driving maneuvres while keeping safe distances to surrounding vehicles, and not on the actual control of the vehicle.
2) Since the algorithm is intended to provide a decision regarding which lane is preferable at each time instance, assuming a lane change reference trajectory exists, the lateral control i.e. steering of the vehicle, can be excluded from the model, within the scope of the research question that is being addressed here. Hence,the longitudinal motion of the vehicle and the force acting upon it can be modelled as a simple double integrator system.
3) Since the algorithm is intended to provide a decision regarding which lane is preferable at each time instance, assuming a lane change exists within one sampling time T (the same amount as the delay in the hybrid automaton).

The assumptions above allow the simplification of vehicle lane changing model dynamics as follow.

$$\begin{aligned} x_0(t+1) &= x_0(t) + v_0(t)T \\ v_0(t+1) &= v_0(t) + u(t)T \\ q_0(t+1) &= q_0(t) + \sigma(t), \\ \sigma(t) &\in \{-1, 0, 1\} \\ q_0(t) &\in \{0, 1, 2, \dots, n\}, \end{aligned} \tag{1}$$

where u is the acceleration/deceleration control signal, x is the vehicle position, v is the vehicle speed, q the lane index where the vehicle locates. In this formulation, the freeway lanes are labeled from left to right as {1, 2, ..., n}.

The velocity and acceleration are considered to be constrained,

$$v_0(t) \in [v_{min}, v_{max}], \quad u(t) \in [u_{min}, u_{max}], \quad \forall t \in \mathbb{Z}^+ \tag{2}$$

We assume that the state of surrounding vehicles are all measurable. The state variables of the j-th surrounding vehicle is denoted by $[x_j(t),\ v_j(t),\ q_j(t)]$. This is illustrated in Figure 2.

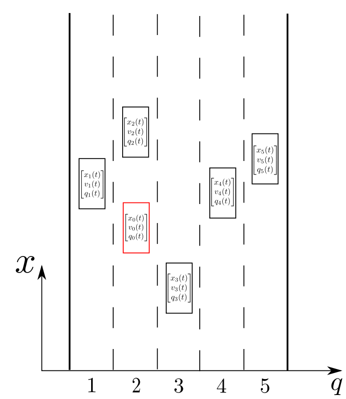

Fig. 2. The environment of the controlled vehicles is assumed to be known

In the next few sections, the aforementioned system will be transformed into an equivalent mixed logical dynamical system that is compatible with the mixed integer programming.

B. Mixed logical dynamical system

The mixed logical dynamical system modeling is a framework that allows a system to combine physical laws, logical rules and operating constraints. A detailed discussion of MLD systems was done by Bemporad *et. al.* [13]. In an attempt to make this paper self-contained, we will only refer to a minimal description about MLD systems required for our work.

As we are interested in systems which have both logic and dynamics, we need to establish a link between the two worlds. As shown in our methodology, we end up with mixed-integer linear inequalities, i.e. linear inequalities involving both continuous variables $x \in \mathbb{R}^n$ and logical variables $\delta \in \{0,1\}$. Consider a bounded linear function $f(x) : \mathbb{R}^n \mapsto \mathbb{R}$, where $f(x)$ has known over (under)-estimates M(or m), it is easy to verify that

$$[f(x) \le 0] \leftrightarrow [\delta = 1] \text{ is true iff } \begin{cases} f(x) \le M(1-\delta) \\ f(x) \ge \epsilon + (m-\epsilon)\delta \end{cases} \tag{3}$$

where $\leftrightarrow$ represents "if and only if", $\epsilon > 0$ is a small tolerance (typically the machine precision). The implication $\rightarrow$ relation between two statements can be shown to satisfy,

$$X_1 \rightarrow \sim X_2 \text{ is equivalent to } \delta_1 + \delta_2 \le 1 \tag{4}$$

where X_i = true is represented by $\delta_i = 1$, $i = \{1,2\}$ and $\sim$ is "not". When three logical variables are related, we have the following equivalence,

$$[\delta_3 = 1] \leftrightarrow [\delta_1 = 1 \wedge \delta_2 = 1]$$

$$\text{is equivalent to } \begin{cases} -\delta_1 + \delta_3 \le 0 \\ -\delta_2 + \delta_3 \le 0 \\ \delta_1 + \delta_2 - \delta_3 \le 1 \end{cases} \tag{5}$$

where $\wedge$ represents "and" logic.

C. Conversion of discrete variables to logical variables

To express the discrete state variable $q = \{1,\dots,n\}$ in terms of logical variables, we define n logical variables $\delta_i = \{0,1\}$ where $i = \{1,\dots,n\}$ as follows,

$$[q = k] \leftrightarrow [\delta_k = 1 \wedge \delta_j = 0, j \ne k], \quad k, j \in \{1,\dots,n\} \tag{6}$$

The conversion should be clear in Table I,

TABLE I
ILLUSTRATION OF THE CONVERSION FROM THE DISCRETE VARIABLE TO MULTIPLE LOGICAL VARIABLES

q	δ_1	δ_2	δ_3	δ_4	δ_5
1	1	0	0	0	0
2	0	1	0	0	0
3	0	0	1	0	0
4	0	0	0	1	0
5	0	0	0	0	1

Logically, we first introduce $\bar{\delta}_k$ and $\underline{\delta}_k$ as indicators of the value of q,

$$\begin{aligned} [q \le k] &\leftrightarrow [\bar{\delta}_k = 1] \\ [q \ge k] &\leftrightarrow [\underline{\delta}_k = 1] \end{aligned} \tag{7}$$

then it should be readily seen that δ_k needs to satisfy the following logic,

$$[\delta_k = 1] \leftrightarrow [\bar{\delta}_k = 1 \wedge \underline{\delta}_k = 1] \tag{8}$$

The aforementioned logic statements must be expressed in terms of a system of inequalities so that we can utilize mixed-integer programming. The inequalities can be derived to be,

$$\begin{cases} q - k \le M_1(1 - \bar{\delta}_k) \\ q - k \ge \epsilon + (m_1 - \epsilon)\bar{\delta}_k \\ -q + k \le M_1(1 - \underline{q}_k) \\ -q + k \ge \epsilon + (m_1 - \epsilon)\underline{\delta}_k \\ -\bar{\delta}_k + \delta_k \le 0 \\ -\underline{\delta}_k + \delta_k \le 0 \\ \bar{\delta}_k + \underline{\delta}_k - \delta_k \le 1 \end{cases} \tag{9}$$

where $k \in \{1,\dots,n\}$, M and m are the upper and lower bound of $q-k$ and $-q+k$. It suffices to use $M_1 = n = -m_1$.

The discrete input $\sigma \in \{-1,0,1\}$ can be written as a linear combination of three logical variables $\sigma_{-1}, \sigma_0, \sigma_1 \in \{0,1\}$,

$$\sigma = 1 \cdot \sigma_1 + 0 \cdot \sigma_0 + (-1) \cdot \sigma_{-1} \tag{10}$$

Their relationship can also be seen from Table II,

TABLE II
ILLUSTRATION OF THE CONVERSION FROM THE DISCRETE INPUT TO THREE LOGICAL VARIABLES

σ	σ_1	σ_0	σ_{-1}
1	1	0	0
0	0	1	0
-1	0	0	1

D. Encoding Safety constraints in linear inequalities

The safety constraint considered in this work is that, when the controlled vehicle and a surrounding vehicle are occupying the same lane, their center-to-center distance has to be greater than a prescribed safety region. The contrapositive of this statement can be translated to the following implication logic,

$$\left(x_k^i - x_k, v_k^i - v_k\right) \in \mathcal{S}, \quad \forall i \in \mathcal{I}_k^{\text{align}} \subseteqq \mathcal{I} \tag{11}$$

$\mathcal{I}_k^{\text{align}} = \left\{i \in \mathcal{I} | L_k^i = L_k\right\}$ is the subset of all neighboring vehicles that are on the same lane as the host vehicle. The safety constraint implies that the lane indicies of the vehicle are confined by the number of lanes, and equation (11) defines the safety constraints. In particular, $\mathcal{S}$ is the safety set, and $\mathcal{I}$ is defined as the set of all neighboring cars. In this work, the safety constraint set is defined as follows,

$$\mathcal{S} = \begin{cases} x_k^i - x_k \ge x_{\text{safe}} \text{ and } v_k^i - v_k \ge -\alpha(x_k^i - x_k) \\ \qquad \text{if } \ x_k^i - x_k \ge 0 \\ x_k^i - x_k \le -x_{\text{safe}} \text{ and } v_k^i - v_k \le \alpha(x_k^i - x_k) \\ \qquad \text{if } \ x_k^i - x_k \le 0 \end{cases} \tag{12}$$

$$\forall i \in \mathcal{I}_k^{\text{align}}.$$

where α is a positive constant defining the braking effectiveness, and x_{safe} is the minimum required distance between vehicles to avoid collisions. The safety constraint (12) can be illustrated as shown in Figure 3.

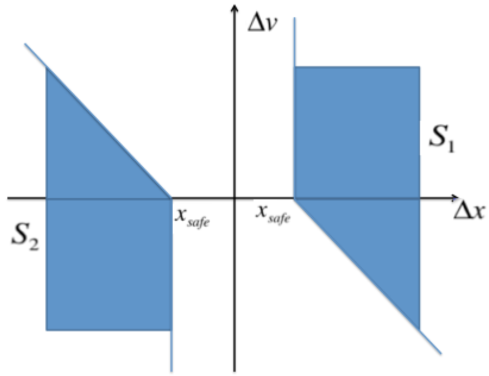

Fig. 3. Safety constraints illustration

The x axis in Figure 3 is the distance between vehicle i and the host vehicle ($x_k^i - x_k$), and y axis is the velocity difference ($v_k^i - v_k$). The safety region is a union of two convex sets S_1 and S_2, where S_1 is the safety region for vehicles ahead of the host vehicle, and S_2 is for vehicles behind.

1) Transforming lane constraint: In order to reformulate of safety constraints (11)-(12) into mixed integer linear constraints, binary variables $\sigma_k^{i,head}$ and $\sigma_k^{i,tail}$ are introduced to indicate if the i^{th} neighboring vehicle is ahead of or behind the host vehicle.

$$x_k^i \geq x_k \leftrightarrow \sigma_k^{i,head} = 1, \forall i \in \mathcal{I}, k = 0, 1, \ldots, N, \tag{13a}$$
$$x_k^i \leq x_k \leftrightarrow \sigma_k^{i,tail} = 1, \forall i \in \mathcal{I}, k = 0, 1, \ldots, N, \tag{13b}$$

Constraint (13) is reformulated as follow by applying (3),

$$x_k^i - x_k \leq \Delta x_{max}(1 - \sigma_k^{i,head}), \tag{14a}$$
$$x_k^i - x_k \geq \epsilon + (-\Delta x_{max} - \epsilon)\sigma_k^{i,head}, \tag{14b}$$
$$x_k - x_k^i \leq \Delta x_{max}(1 - \sigma_k^{i,tail}), \tag{14c}$$
$$x_k - x_k^i \geq \epsilon + (-\Delta x_{max} - \epsilon)\sigma_k^{i,tail}, \tag{14d}$$
$$\forall i \in \mathcal{I}, k = 0, 1, \ldots, N,$$

where Δx_{max} is the furthest distance of the vehicle that the host car can sense and measure.

2) Safety constraint in mixed-integer inequalities: Then the safety constraints (11) are equivalent to

$$\sigma_k^{i,head} \wedge \sigma_k^{i,(l)} \rightarrow \begin{cases} x_k^i - x_k \geq x_{\text{safe}} \\ v_k^i - v_k \geq -\alpha(x_k^i - x_k) \end{cases} \tag{15a}$$
$$\sigma_k^{i,tail} \wedge \sigma_k^{i,(l)} \rightarrow \begin{cases} x_k^i - x_k \leq -x_{\text{safe}} \\ v_k^i - v_k \leq \alpha(x_k^i - x_k) \end{cases} \tag{15b}$$
$$\forall k = 0, 1, \ldots, N, \forall i \in \mathcal{I}.$$

In order to transform safety constraints (15) to linear mix integer constraint, we introduce another set of binary variables σ_k^{i,S_0}, and σ_k^{i,S_1} that satisfy

$$\sigma_k^{i,S_0} \leftrightarrow \begin{cases} x_k^i - x_k \geq x_{\text{safe}} \\ v_k^i - v_k \geq -\alpha(x_k^i - x_k) \end{cases} \tag{16a}$$
$$\sigma_k^{i,S_1} \leftrightarrow \begin{cases} x_k^i - x_k \leq -x_{\text{safe}} \\ v_k^i - v_k \leq \alpha(x_k^i - x_k) \end{cases} \tag{16b}$$
$$\forall k = 0, 1, \ldots, N, \forall i \in \mathcal{I}.$$

Therefore, the safety constraint (15) can be rewritten as

$$\sigma_k^{i,head} \wedge \sigma_k^{i,(l)} \rightarrow \sigma_k^{i,S_0} \tag{17a}$$
$$\sigma_k^{i,tail} \wedge \sigma_k^{i,(l)} \rightarrow \sigma_k^{i,S_1} \tag{17b}$$

Finally logic constraints (17) are reformulated as linear mix integer constraints

$$2 - (\sigma_k^{i,head} + \sigma_k^{i,(l)}) \geq \epsilon - \epsilon\sigma_k^{i,S_0} \tag{18a}$$
$$2 - (\sigma_k^{i,tail} + \sigma_k^{i,(l)}) \geq \epsilon - \epsilon\sigma_k^{i,S_1} \tag{18b}$$

E. Mixed integer programming

Now, the optimal controller synthesis problem is readily to be cast into a mixed-integer (linear) programming formulation for optimization. The optimality is defined so that the controlled vehicle has the fastest allowed velocity while undergoing the minimal number of lane changing maneuvers. Hence, the mixed integer programming problem is written as,

$$\min_{u(\bullet),\sigma(\bullet)} \underbrace{-\sum_{k=0}^{N} v_0(k)}_{\text{encourage fast speed}} \underbrace{-\sum_{k=0}^{N-1} \sigma_0(k)}_{\text{discourage lane changing}}$$

subject to

continuous state dynamics: Eqn. (1)

continuous state and input constraints: Eqn. (2)

discrete state constraint: Eqn. (9)

discrete input constraint: Eqn. (10)

initial condition: $[x_0(0),\ v_0(0),\ q_0(0)]^T$ is current state

safety constraint: Eqn. (18) (for all surrounding vehicles) (19)

Note that in the safety constraints, the state variables of all surrounding vehicles $[x_j(t),\ \ v_j(t),\ \ q_j(t)]$ need to be predicted. In our implementation, the assumptions of zero acceleration and no lane change have been made. To account for the uncertainties of surrounding vehicles, the idea of the moving horizon method (model predictive control) can be utilized. In other words, after solving the optimization problem and obtaining the optimal control sequence $\{u^*(0), \ldots, u^*(N-1), \sigma^*(0), \ldots, \sigma^*(N-1)\}$, we only apply the first control input $[u^*(0),\ \sigma^*(0)]^T$, and solve the optimization again with the shifted time index and the new environment.

III. Simulation Results

The above stated formulation of mixed integer linear programming is implemented in Matlab using *Yalmip* [14] and IBM CPLEX solver [15]. A five-lane freeway is simulated. The model of continuous state dynamics is discretized with a sampling time interval of $T = 5$ sec. The speed and acceleration constraints are assumed to be $[25, 35](m/s)$ and $[-5, 5](m/s^2)$ respectively. These numbers are only adopted in a test case for illustration, and these numbers can certainly be adjusted to match closely with various operating scenarios, There are six vehicles under consideration, one controlled vehicle (in red) and 5

TABLE III

INITIAL CONDITIONS USED IN THE SIMULATION TWO

Index j	$x_j(0)(m)$	$v_j(0)(m/s)$	$q_j(0)$
ego	0	35	3
1	20	25	3
2	30	30	4
3	15	32	2
4	20	28	5
5	20	30	1

surrounding vehicles (in black).

We have assumed the initial conditions, which are listed in Table III to demonstrate the effectiveness of our proposed methodology.The ego vehicle is approaching the slower moving, while right lane vehicle has sufficient large distances to allow for lane changing but sorrounding vehicle is approaching at high speed, so the ego vehicle would change lane twice. The detail manoeuvres are shown in Figure 4 and 5.

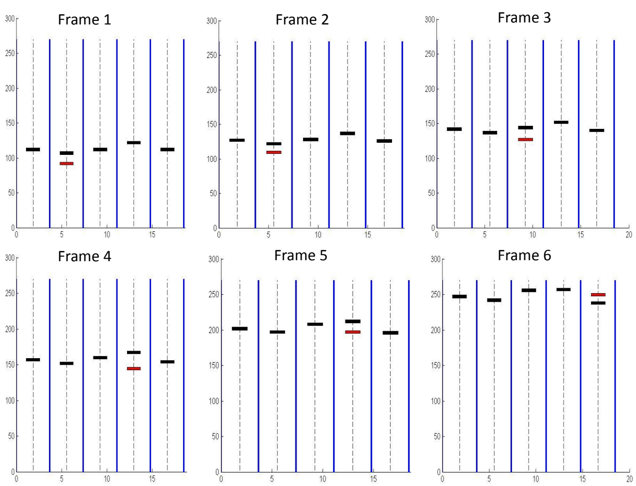

Fig. 4. Lang changing animation one

From the simulation result shown in Figure 6 and 7, which intial condition is presented by Table IV, we observe that the controlled vehicle changes to the fourth lane, then follows the vehicle in that lane which is relatively faster than other surrounding vehicles and finally changes to the rightmost lane when the space opens up. During the simulation, the controlled vehicle gradually increased its speed while maintaining safety zones around it. This lane changing maneuver confirms the validity of our methodology.

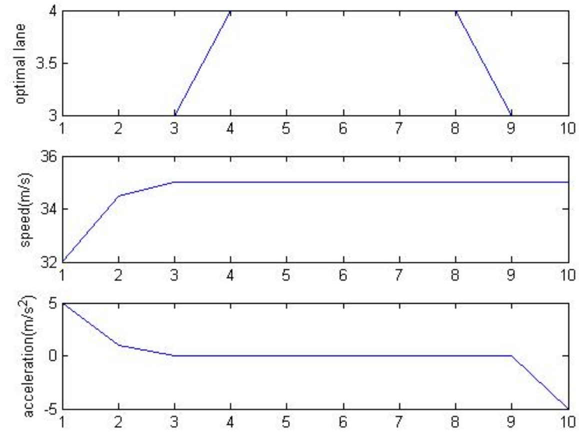

Fig. 5. Lang changing simulation one-optimal result

TABLE IV

INITIAL CONDITIONS USED IN THE SIMULATION ONE

Index j	$x_j(0)(m)$	$v_j(0)(m/s)$	$q_j(0)$
ego	0	30	2
1	20	31	3
2	30	30	4
3	15	28	2
4	20	28	5
5	20	30	1

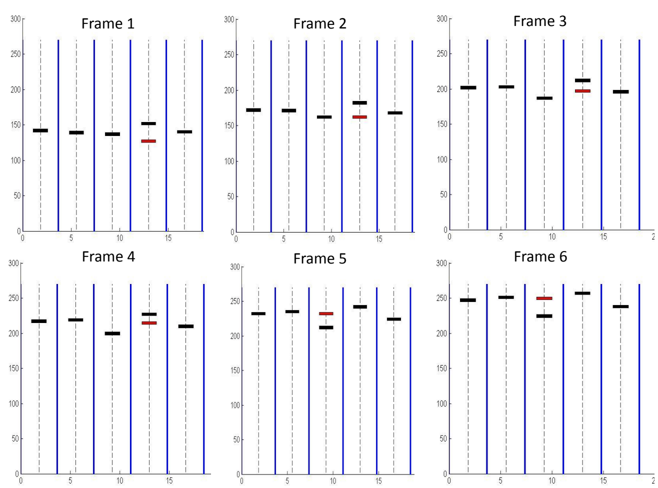

Fig. 6. Lang changing animation two

From the figures it can be seen that the collision avoidance constraints ensure that ego vehicle maintains safe distances to the relevant surrounding vehicles by allowing to increase speed.In the figure it can be seen that the algorithm generates a control signal for the lateral movement that allows the ego vehicle to maintain its desired velocity if possible, or else adjusts it to the velocity of an appropriate surrounding vehicle.

The animation snapshots shown here are from a single optimization. The moving horizon strategy has not been provided here due to page limitation and scope of work for this paper, but the extension should be straightforward.

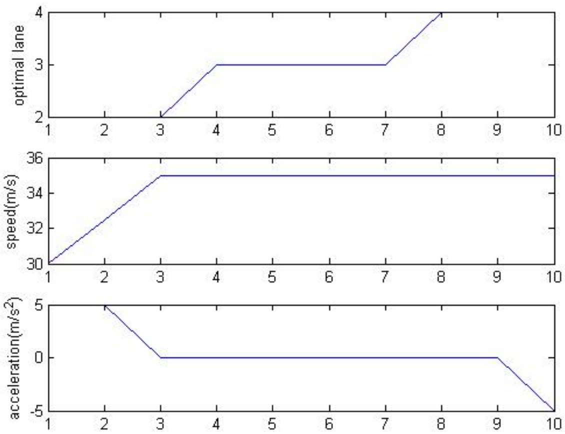

Fig. 7. Lang changing simulation two-optimal result

IV. CONCLUSIONS

This paper present a novel decision and control algorithm of decision making strategies for lane change and overtake maneuvres.

By reducing the complexity of the system model and introducing a binary decision variable, a model predictive controller is efficiently computed. The predictive controller allows full control of acceleration/deceleration as well as providing a decision variable regarding preferred lane at each time instance. Simulation results demonstrate and confirm the flexibility and capability of the proposed algorithm to make decisions and control actions in a manner similar to human driving behaviour i.e. maintaining a desired velocity while preventing intervehicle distances to become unsafely short. The illustrated case represents a driving environment of mutiple-lane, one-way freeway traffic scenario. These results motivate further work in refining the algorithm to incorporate a prediction model for the dynamic behaviour of surrounding vehicles,which no longer assumes that surrounding vehicles move at constant velocity without lane change maneuvers of theris own. The model can be further extended to include uncertainties and noise in the measured sensor information.

For the future studies, our reseach topics will involve the development of an overall lane changing controller by combing the optimal approach and dynamic game approaching. Investigation is also warranted to address concerns that (1) mixed integer programming suffers from combinatorial complexity and (2) the required computational time is strongly influenced by the number of binary variables included in the problem formulation. These issues are crucial for the optimization process in embedded systems for vehicle control.

REFERENCES

[1] Luke Fletcher, Seth Teller, Edwin Olson, David Moore, Yoshiaki Kuwata, Jonathan How, John Leonard, Isaac Miller, Mark Campbell, Dan Huttenlocher, Aaron Nathan, and Frank-Robert Kline. The mit – cornell collision and why it happened. In Martin Buehler, Karl Iagnemma, and Sanjiv Singh, editors, *The DARPA Urban Challenge*, volume 56 of *Springer Tracts in Advanced Robotics*, pages 509–548. Springer Berlin Heidelberg, 2009.

[2] A. Furda and L. Vlacic. Enabling safe autonomous driving in real-world city traffic using multiple criteria decision making. *Intelligent Transportation Systems Magazine, IEEE*, 3(1):4–17, Spring 2011.

[3] Junqing Wei, J.M. Dolan, and B. Litkouhi. A prediction- and cost function-based algorithm for robust autonomous freeway driving. In *Intelligent Vehicles Symposium (IV), 2010 IEEE*, pages 512–517, June 2010.

[4] D. C K Ngai and N. H C Yung. A multiple-goal reinforcement learning method for complex vehicle overtaking maneuvers. *Intelligent Transportation Systems, IEEE Transactions on*, 12(2):509–522, June 2011.

[5] DianBo Ren, JiYe Zhang, JingMing Zhang, and ShengMin Cui. Trajectory planning and yaw rate tracking control for lane changing of intelligent vehicle on curved road. *Science China Technological Sciences*, 54(3):630–642, 2011.

[6] K.A. Redmill C.Hatipoglu, U.Ozguner. Automated lane change controller design. *IEEE Transactions on Intelligent Transportation Systems*, 4(1):13–22, March 2003.

[7] Chaozhong Wu, Zhen Huang, R. R. Negenborn, and Huarong Zheng. *Model Predictive Control for Intelligent Vehicle Lane Change*, chapter 37, pages 265–276.

[8] J. Nilsson and J. Sjoberg. Strategic decision making for automated driving on two-lane, one way roads using model predictive control. *Intelligent Vehicles Symposium (IV), 2013 IEEE*, pages 1253–1258, June 2013.

[9] M. Mukai and T. Kawabe. Model predictive control for lane change decision assist system using hybrid system representation. *SICE-ICASE, 2006. International Joint Conference*, pages 5120–5125, Oct 2006.

[10] R. Schubert. Evaluating the utility of driving: Toward automated decision making under uncertainty. *Intelligent Transportation Systems, IEEE Transactions on*, 13(1):354–364, March 2012.

[11] M. Ardelt, C. Coester, and N. Kaempchen. Highly automated driving on freeways in real traffic using a probabilistic framework. *Intelligent Transportation Systems, IEEE Transactions on*, 13(4):1576–1585, December 2012.

[12] J.E. Naranjo, C. Gonzalez, R. Garcia, and T. de Pedro. Lane-change fuzzy control in autonomous vehicles for the overtaking maneuver. *Intelligent Transportation Systems, IEEE Transactions on*, 9(3):438–450, September 2008.

[13] Alberto Bemporad and Manfred Morari. Control of systems integrating logic, dynamics, and constraints. *Automatica*, 35(3):407–427, 1999.

[14] Johan Lofberg. Yalmip: A toolbox for modeling and optimization in matlab. In *Computer Aided Control Systems Design, 2004 IEEE International Symposium on*, pages 284–289. IEEE, 2004.

[15] IBM ILOG CPLEX. V12. 1: User's manual for cplex. *International Business Machines Corporation*, 46(53):157, 2009.