Optimized Speed Trajectories for Cyclists, Based on Personal Preferences and Traffic Light Information-A Stochastic Dynamic Programming Approach

Dabiri, Azita; Hegyi, Andreas; Hoogendoorn, Serge

DOI
10.1109/TITS.2020.3014448

Publication date
2022

Document Version
Final published version

Published in
IEEE Transactions on Intelligent Transportation Systems

Optimized Speed Trajectories for Cyclists, Based on Personal Preferences and Traffic Light Information—A Stochastic Dynamic Programming Approach

Azita Dabiri, Andreas Hegyi, *Member, IEEE*, and Serge Hoogendoorn

***Abstract*— The literature on green mobility and eco-driving in urban areas has burgeoned in recent years, with special attention to using infrastructure to vehicle (I2V) communications to obtain optimal speed trajectory which minimize the economic and environmental costs. This article shares the concept with these studies but turns the spotlight on cyclists. It examines the problem of finding optimal speed trajectory for a cyclist in signalised urban areas. Unlike the available studies on motorised vehicles which predominantly designed for pre-defined, fixed traffic lights timing, this article uses an algorithm based on stochastic dynamic programming to explicitly address uncertainty in traffic light timing. Moreover, through a comprehensive set of simulation experiments, the article examines the impact of the speed advice's starting point as well as the cyclist's willingness for changing his/her speed on enhancing the performance. The proposed approach targets various performance metrics such as minimising the total travel time, energy consumption, or the probability of stopping at a red light. Hence, the resulting speed advice can be tailored according to the personal preferences of each cyclist. In a simulation case study, the results of the proposed approach is also compared with an existing approach in the literature.**

***Index Terms*— Speed advice, cycling, energy consumption, travel time, stochastic dynamic programming.**

I. Introduction

TIME loss, stress, and air and sound pollution are only a few items of the long list of adverse effects caused by the abundant use of motorised vehicles in big cities. By choosing bikes over motorised vehicles, in addition to health benefits, citizens can contribute in mitigating these effects. Obviously, this is a great incentive for policy makers to look for ways that raise public interest for cycling.

The level of convenience that the users of motorised vehicles can benefit from, is a strong motivation for them to favour car over bicycle for their daily commuting. An approach to shift the public interest in favour of cycling is to introduce policies that make cycling more appealing. Such policies are followed by policy makers in for instance the Netherlands, Denmark and Germany and include provision of cycling facilities, careful integration with public transport, and assigning expensive toll and parking fees for cars in city centers [1]. That being said, measures that assist cyclists to choose the route and speed profiles that meet their cycling preferences have no less significance [2].

The technological advances have opened up a wide range of design possibilities for providing assistance to cyclists. In most of these designs, computer acts as a passive player and the assistance is limited to a form of advice in e.g. navigation or change in speed. The advice may be given through the rider's mobile display [3], a display on the bike's handlebar [4], a projection on the road surface [5] or even through an audio assistance connected to a smart cycling insole [6]. With increasing the popularity of electric bikes (ebikes), more variation of assistance systems have been designed that are not passive and by exploiting the actuation possibilities may provide a physical assistance to the riders through adjusting the engine of the ebike [7]–[9]. Understanding the cyclists' perception when having this interactive experience with a smart bike as such is extensively addressed in [7]. In light of these recent development in the design of suitable human-machine interface for cyclists, this article aims at developing an algorithm that gives speed advice to the cyclist while optimising the metric, or possibly combination of various metrics, that represent the rider's interest when cycling in signalised urban areas.

Research shows that in their daily trips, cyclists may detour the shortest path and choose a longer path with fewer signalised intersections [10]. Without any advice, cyclists may make wrong decisions while approaching an intersection due to their incomplete knowledge of the traffic light timing. Such decisions may result in waste of time and energy. For example, when the cyclist sees green light at the junction, he/she usually accelerates, hoping to reach the intersection while the light is still green. If he/she fails, the cyclist has used unnecessary energy without passing the intersection. Moreover, full stop at junctions due to red light is undesirable for some of the riders, especially the elderly, as it is more difficult for them to get off and on the bike if they stop for the red light. Riders also need to use extra energy to reach the speed they had before their stop at the intersection. In cities like Copenhagen

Manuscript received July 18, 2019; revised April 4, 2020 and July 17, 2020; accepted July 30, 2020. Date of publication September 14, 2020; date of current version February 2, 2022. This work was supported by the Amsterdam Institute for Advanced Metropolitan Solutions (AMS). The Associate Editor for this article was A. Nunez. *(Corresponding author: Azita Dabiri.)*

The authors are with the Department of Transport and Planning, Delft University of Technology, 2628 Delft, The Netherlands (e-mail: a.dabiri@tudelft.nl; a.hegyi@tudelft.nl; s.p.hoogendoorn@tudelft.nl).

Digital Object Identifier 10.1109/TITS.2020.3014448

and Amsterdam, the timing of some traffic lights is designed in favour of cyclists. With these approaches, if cyclists keep their speed at a certain value, they can enjoy a green-wave while passing multiple intersections [1]. However, favouring bikes over motorised vehicles is not always possible and can raise some concerns, as it may contribute in higher congestion of vehicles. Moreover, creating green waves for cyclists is not possible everywhere since it may conflict with public transport priorities in cities.

One way to proceed is to use the traffic light phase and timing information and advise the cyclists to adjust their speed accordingly, in order to increase their chance of catching green or to optimize for other personal preferences. Although the research on speed advisory systems for cyclists in signalised areas is limited [7], [11], [12], algorithms and methods to find the optimal trajectory for motorised vehicles in signalised urban areas have been extensively explored in the literature. These approaches may differ due to e.g. their difference in the utilised control method, the control objectives, the type of vehicles, the level of vehicle's connectivity, and the level of autonomy of vehicles considered in the problem. The problem is addressed for single motorised vehicle by using e.g. model predictive control [13]–[15], dynamic programming [16], [17] or other optimization-based algorithms [18], [19]. Various driver assistant systems have been also field tested in [20]–[22]. More recently, driver assistant systems for connected and autonomous vehicles and mixed traffic are studied as well [23]–[29]. A platoon-based strategy for cooperation between human-driven vehicles and automated vehicles is developed in [27] and model predictive control is deployed to minimise the platoons' fuel consumption. In [28], platoons are reorganised to maximise the number of vehicles that pass the intersection during the current green phase while safety, passenger comfort and fuel consumption are considered. Connected vehicles are grouped based on their estimated arrival time to the intersection into different clusters in [29] where eco-driving is achieved by the cooperation between the leader of each cluster with the leaders of the platoons within the cluster.

Due to the increasing market penetration of electrical vehicles (EV) and hybrid electrical vehicles (HEV), studies on the energy efficiency of these types of vehicles in signalised urban areas have gained a lot of attention in recent years as well. In [30], the green light optimal speed advisory system (GLOSA) algorithm is combined with a fuel consumption map of a HEV and genetic algorithm is used to find the optimal speed for HEV that minimize the total equivalent fuel consumption of the vehicle while passing the intersections. An algorithm is suggested in [31] that in addition to traffic light state, vehicle queues at the intersection are communicated to HEVs in order to plan the vehicle trajectory that minimize fuel consumption and keep the jerk value of the vehicles low. A multi-stage optimisation algorithm is formulated in [32] to optimise the overall energy consumption for an EV traveling through a corridor consisting of multiple signalized intersections while taking into account the state of the traffic light and also the state of vehicles queue at each intersection. Model predictive control is used in [14] to achieve eco-driving in a HEV using traffic signal information.

All of the aforementioned methods, for single vehicle and connected vehicles, are based on the presumption that the traffic light at the intersection follows a fixed-time regime. An available alternative in the literature that takes the stochastic nature of traffic signal timing into account is [16], where the probability of green in the future is predicted based on the current colour of the traffic light and the average timing data. This probability prediction is embedded in a cost function and deterministic dynamic programming is used to find the optimal speed advice. In this article, a method based on stochastic dynamic programming is suggested that improves the results of [16] and is also applicable for traffic lights with variable cycle time.

The unique contribution of this article is the use of stochastic dynamic programming to provide speed advice for cyclists, taking into account the stochastic nature of traffic light timing and also different personal preferences that a cyclist has. The developed algorithm in this article is a general framework and can be used for various junction signalisation types, varying from fixed-time to traffic-responsive controllers. It extends our preliminary results presented in [11] on three main aspects. First, the phase transition is extended by including amber phase in addition to red and green phases, which make the simulation more close to the reality. Second, the optimal speed advice generated by the proposed algorithm for various preferences of cyclists is compared in multiple illustrative case studies. Third, the simulation case studies are complemented by comparison between the proposed method and the approach suggested in [16].

The rest of the article is organised as follows. In Section II, kinematics of a cyclist and dynamics of the traffic light phase and timing are explained. Cyclist preferences and the corresponding reward function are detailed in Section III, which is followed by problem formulation in Section IV. For solving the optimization problem a stochastic dynamic programming approach is adopted, which is detailed in Section V. Comprehensive simulation case studies are presented in Section VI. Section VII discusses the implementation and future research directions, and Section VIII highlights this article's key conclusions.

II. Process Description

This section is focused on the modelling of the longitudinal movement of a cyclist and its environment. By the environment, we refer to the control structure and timing of the traffic light towards which the cyclist is moving. While the kinematics of the bike follows a set of deterministic rules, as explained in the following, we allow uncertainty in the evolution of the traffic light state. Such formulation allows to describe a wide variety of traffic signal timing types, including the fixed-time and the traffic-responsive controllers.

A. Cyclist's Kinematics

We use a discrete-time description with the discretisation time Δt (s) and the corresponding counter k, denoting the

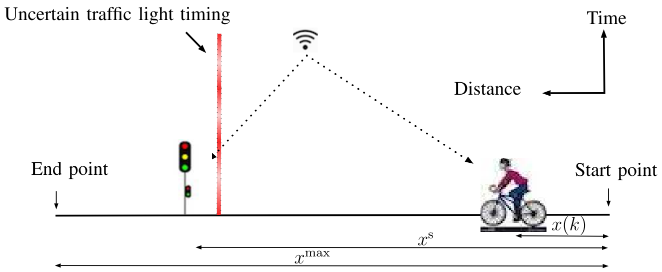


Fig. 1. Speed advice for cyclist moving towards an intersection with uncertain signal timing.

time $t = k\Delta t$, to describe the movement of a cyclist depicted in Figure 1. In time step k, the bike is characterised by its speed $v(k)$ (m/s), and its position $x(k)$ (m) with respect to the upstream end of the road that leads to the intersection. Hence, we can describe the kinematics of the bike by the following set of discrete-time equations:

$$\begin{aligned} x(k+1) &= x(k) + v(k)\Delta t + \frac{1}{2}u(k)\Delta t^2, \\ v(k+1) &= v(k) + u(k)\Delta t, \end{aligned} \tag{1}$$

where $u(k)$ (m/s^2) is the acceleration of the bike that is bounded as:

$$u^{\min} \le u(k) \le u^{\max}. \tag{2}$$

Let us define the cyclist's state in time step k as $S_k^{\mathrm{c}} \in \mathcal{S}^{\mathrm{c}}$ where

$$\mathcal{S}^{\mathrm{c}} = \{(v, x) | 0 \le v \le v^{\max}, 0 \le x \le x^{\max}\} \tag{3}$$

and

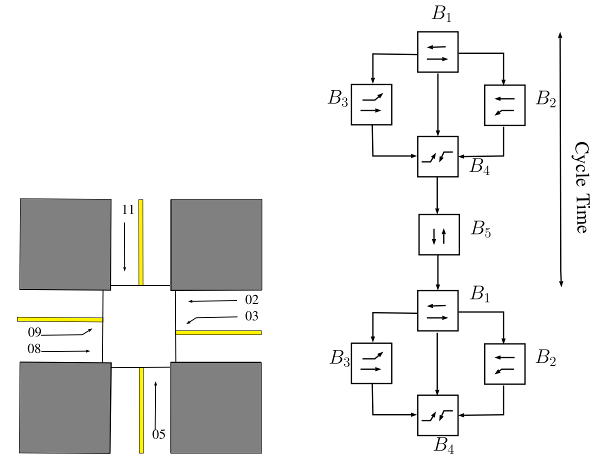
$$S_k^{\mathrm{c}} = \big(v(k), x(k)\big), \tag{4}$$

where $x^{\max}$ is the distance between the start and the end point and $v^{\max}$ denotes the maximum speed of the bike. The state of the cyclist in the time step $k+1$ will evolve according to (1) and as a result of acceleration $u(k)$. As it will be of use in the following sections, let us choose another way to represent (1), by defining the transition probability $P^a_{s_c s_c'}$. With this term, we express the transition probability of going from state s_c to state s_c' by taking the acceleration a and respecting the dynamics in (1):

$$\begin{aligned} P^a_{s_c s_c'} &= \mathbb{P}\,[S^c_{k+1} = s_c' \mid S^c_k = s_c, u(k) = a] \\ &= \begin{cases} 1, & \text{If (1) is respected,} \\ 0, & \text{otherwise} \end{cases} \end{aligned} \tag{5}$$

B. Modeling of the Traffic Light Phase and Timing

An intersection is composed of multiple streams (also called movements), where each stream is defined as a unique combination of an incoming and a leaving travel direction. Depending on these directions, some combination of streams are conflicting, and consequently they are not allowed to have green at the same time. Phase control of traffic lights determines the sequence and the time duration that each stream is assigned to use the conflict area. Take the intersection depicted in Figure 2a as an example. The intersection has 6 streams, coded according to the Dutch standard. For this intersection, a flow diagram such as the one depicted in Figure 2b describes the allowed transitions from one block to another block in one cycle time, where each block in the flow diagram corresponds to a set of non-conflicting streams (in this example each block consists of two streams). Table I enumerates each block with the associated streams and their corresponding light colour. To have a general description, in this article, the sequence of blocks that get served in one cycle time is allowed to be stochastic. In other words, the transition from one block to the other and the time that this transition happens is not deterministic but described by a transition probability. As an example, from Figure 2b, it is clear that B_4 can be reached either via B_1, B_2 or B_3. Depending on whether the stream 8 or 2 ends earlier in block B_1, one of the conflicting streams 3 or 9 can be given green before B_4 starts. In practice, the transition probability of going from one block to the other can be a function of, e.g, traffic demand on the road or queue length. As explained in more detail in the following, in this article, we assume that this transition probability is known from the historical data and is a function of the elapsed time of the streams that are currently green.



(a) Schematic representation of an intersection

(b) Flow diagram

Fig. 2. An intersection and its flow diagram.

TABLE I
BLOCKS AND THE ASSOCIATED STREAM PAIRS

Block	B_1	B_2	B_3	B_4	B_5
Streams' code	(2,8)	(2,3)	(8,9)	(3,9)	(5,11)

In order to explicitly model the yellow lights for the streams of which the green time has passed, we include intermediate blocks. Having the intermediate blocks defined as in Table II, Figure 3 depicts the possible transitions between all the blocks (including the intermediate ones) in one cycle time

TABLE II
INTERMEDIATE BLOCKS AND THE ASSOCIATED STREAMS' PAIRS

Block	B_6	B_7	B_8	B_9	B_{10}	B_{11}	B_{12}
Streams' code	(2,8)	(2,3)	(2,8)	(8,9)	(3,9)	(5,11)	(2,8)

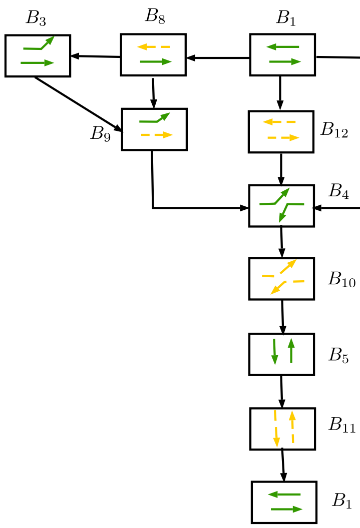


Fig. 3. Main and intermediate block transitions in one cycle time.

for the intersection shown in Figure 2a. The same logic used in Figure 2b can be used to interpret Figure 3.

To put the evolution of signal phase and timing in mathematical terms, let us first define some notations. We use B_b to represent the b-th block and index L to represent the total number of blocks (including the intermediate ones) used to describe the traffic light control structure of the intersection. For instance, L is equal to 12 in the chosen configuration in Figure 3. Moreover, $n_b^j(k)$ for $j \in \{1, \dots J\}$ indicates the elapsed time steps of the light in the stream code corresponding to the j-th stream of B_b at time step k where J is the total number of streams included in B_b. Note that to enforce the maximum green or yellow time for each stream, $n_b^j(k)$ can not exceed $N_{B_b}^j$. If the state $\mathcal{S}^l$ of the traffic light is defined as:

$$\mathcal{S}^l = \left\{(B_b, n_b^1, \dots, n_b^J) \mid B_b \in \mathcal{B},\ n_b^1 \in \mathcal{N}_b^1,\ \dots, n_b^J \in \mathcal{N}_b^J\right\} \tag{6}$$

where,

$$\mathcal{B} = \{B_1, B_2, \dots, B_L\}, \tag{7}$$

$$\mathcal{N}_b^1 = \{1, 2, \dots N_{B_b}^1\}, \quad b \in \{1, \dots L\}, \tag{8}$$

$$\mathcal{N}_b^J = \{1, 2, \dots N_{B_b}^J\}, \quad b \in \{1, \dots L\}, \tag{9}$$

then $S_k^l \in \mathcal{S}^l$ defined as follows to represent the state of the traffic light at time step k:

$$S_k^l = \left(B(k), n_b^1(k), \dots n_b^J(k)\right). \tag{10}$$

Let us use an example to make the (10) more clear. All the blocks in Figure 3 have two streams and hence, J for all of the blocks is equal to 2. Moreover, the triple $s_l = (B_3, 10, 5)$ refers to the state that the pair of streams (8,9) (corresponding to B_3 in Table I) have been green for 10 and 5 time steps respectively.

Given the state definition in (10), we describe the evolution of the traffic signal time and phase by specifying the transition probability $P_{s_l s_l'}$:

$$P_{s_l s_l'} = \mathbb{P}\,[S_{k+1}^l = s_l' \mid S_k^l = s_l], \quad \forall s_l,\ s_l' \in \mathcal{S}^l. \tag{11}$$

Note that the one-step dynamics (11) displays the Markov property of the process as it enables us to predict the next state of the traffic light independently of the past and by only knowing the current state.

C. Model of the Whole Process

Knowing the kinematics of the bike through (5) and the evolution of the traffic signal by (11), let us define the state of the whole process at time step k with

$$S_k = (S_k^l, S_k^c) \in \mathcal{S}, \quad \mathcal{S} = \{(s_l, s_c) \mid s_l \in \mathcal{S}_l, s_c \in \mathcal{S}_c\}. \tag{12}$$

Then the dynamics of the process can be described through the transition probability $P_{ss'}^a$ which is obtained by the following equation:

$$P_{ss'}^a = P_{s_l s_l'} P_{s_c s_c'}^a. \tag{13}$$

Note that the process described above, inherits the Markovian property.

III. CYCLIST PREFERENCES AND OBJECTIVE FUNCTION

Knowing the instantaneous state of the process described in (12), our goal is to recommend the optimal acceleration to each cyclist at each time step k that minimises a metric of their whole trip, such as energy consumption, travel time, probability of stoping due to a red light, or some careful combination thereof. The performance metric may vary for each cyclist as it represents the personal cycling preferences of each rider. Hence, before making a concise problem formulation, we need to categorise the cycling preferences and assign a function to quantify each of them. We should note that it is assumed that there is a platform such as an app such that the cyclist can enter his/her preferences for his/her cycling experience. In general, the cyclist's preferences may be identified from data and using e.g., learning algorithms. However this topic is not in the scope of this article.

To quantify a cyclist's preferences, let us define the following function as the summation of multiple normalised terms, corresponding to various cycling needs and interests. If $S_k = s = (B, n^1, n^2, v, x)$, $u(k) = a$, and $S_{k+1} = s' = (B', n'^1, n'^2, v', x')$, then the quality of the transition from state S_k to S_{k+1} can be quantified by r_{k+1} which is expressed as:

$$\begin{aligned} r_{k+1} = {} & W^{\mathrm{f}} F^{\mathrm{f}}(s, s', a) + W^{\mathrm{i}} F^{\mathrm{i}}(s) \\ & + W^{\mathrm{c}} F^{\mathrm{c}}(s, s', a) + W^{\mathrm{d}} F^{\mathrm{d}}(s, a) + W^{\mathrm{s}} F^{\mathrm{s}}(s, s', a) \\ & + W^{\mathrm{t}} F^{\mathrm{t}}(s) + W^{\mathrm{e}} F^{\mathrm{e}}(s, a), \end{aligned} \tag{14}$$

where W^*, $* \in \{\mathrm{f, i, c, d, s, t, e}\}$ is used to assign a proper weight for the terms in (14), which correspond to safety, avoiding instability, having smooth moneouver, cycling at the desired speed, avoiding stop, travel time, and energy consumption respectively. Thanks to these weights, it is possible to carefully express tradeoffs between the terms in (14). The higher value of r_{k+1} expresses a more desired transition cost. The remainder of this section presents each term in (14) in more detail. In the following, in addition to s, a, and s' defined earlier, we make use of two terms namely x^{s} and B^{s} to respectively denote the location of the stop light and the set of all blocks in Tables I and II that don't contain green for the stream that the cyclist will use to cross the intersection.

1) Safety: Clearly the advice given to the cyclist should not result in unsafe manoeuvres like passing the red light. To prevent such advices, function $F^{\mathrm{f}}(s, s', a)$ is defined as follows to assign a negative value $-R^{\mathrm{f}} < 0$ as a penalty for actions that results in unsafe manoeuvres:

$$F^{\mathrm{f}}(s, s', a) = \begin{cases} -R^{\mathrm{f}}, & \text{if } (s, s') \in \mathcal{S}^f \\ 0, & \text{else} \end{cases} \tag{15}$$

where

$$\mathcal{S}^{\mathrm{f}} = \{(s, s') |\, x \leq x^{\mathrm{s}}, x' > x^{\mathrm{s}}, B \in B^s\} \\ \cup \{(s, s')| x < x^{\mathrm{s}}, x' = x^{\mathrm{s}}, B' \in B^s\} \tag{16}$$

Note that in (16), the first term corresponds to red light running though we do not penalise if the light gets yellow while passing the intersection. Moreover, through the second term, we penalise the hazardous situations of being exactly at the stop line when the traffic light is red.

2) Avoiding Instability: Cyclists may feel unstable if they cycle in a very low speed. Hence, by use of a speed-dependent function $F^{\mathrm{i}}(s')$, the algorithm should discourage actions leading to such situation. $F^{\mathrm{i}}(s')$ should be chosen to have respectively higher absolute value in lower speed. We have selected the following function for $F^{\mathrm{i}}(s')$ in this article where κ is a constant parameter and v^{i} denotes the minimum speed that the cyclist feels stable when cycling.

$$F^{\mathrm{i}}(s') = \begin{cases} \dfrac{-\kappa}{v' + \kappa}, & \text{if } \ 0 < v' < v^{\mathrm{i}} \\ 0, & \text{else} \end{cases} \tag{17}$$

3) Smooth Manoeuvre: To reflect on the user's comfort, the advice given to the cyclist must avoid severe deceleration or harsh acceleration. For that purpose, if u^{max} denotes the maximum possible acceleration, F^c is introduced in the reward function (14) to penalise an action that its resulting speed trajectory is not smooth:

$$F^{\mathrm{c}}(s, s', a) = -\frac{(v - v')^2}{(u^{\max}\Delta t)^2}. \tag{18}$$

4) Cycling at the Desired Speed: Cyclist find it uncomfortable if the speed advice differs very much from his/her desired speed v^{d}. Function $F^d(s, a)$ imposes penalty when such deviation happens. A possible choice for $F^d(s, a)$ is the quadratic function defined as follows:

$$F^{\mathrm{d}}(s, a) = -\frac{(v + a\Delta t - v^{\mathrm{d}})^2}{\alpha}, \tag{19}$$

where α is used for the normalisation and is defined as:

$$\alpha = \max\big((v^{\mathrm{d}})^2, (v^{\max} - v^{\mathrm{d}})^2\big). \tag{20}$$

5) Avoiding Any Stop: As it is mentioned earlier, stopping in front of the red light is one of the sources of frustration for some cyclists. The annoyance is not limited to the need for getting off and on the bike for each stop. Cyclists may also find it inconvenient when they have to use a lot of power to reach the speed they had before they needed to stop at the junction. Function $F^s(s, s', a)$ defined as follows penalises the advice that results in a stop for the cyclist, including stoping at the red light:

$$F^{\mathrm{s}}(s, s', a) = \begin{cases} -R^s, & \text{if } (s, s') \in \mathcal{S}^s \\ 0, & \text{else} \end{cases} \tag{21}$$

where,

$$\mathcal{S}^{\mathrm{s}} = \{(s, s') |\, x = x'\}, \tag{22}$$

and $-R^{\mathrm{s}} < 0$ is a constant negative number.

6) Minimising Total Travel Time: A gain in the travel time for a cyclist is expected when the cyclist can travel faster (than normally) at the right moments, e.g., just before the traffic light gets red, to catch the end of green, or just before the end of red, when the rider knows that he/she does not have to slow down, or can accelerate in advance because the light will be green when he/she reaches to the intersection. To incorporate the travel time in the optimisation problem, function F^t is introduced as follows to imposes penalty with constant negative number $-R^t$ at every time step:

$$F^{\mathrm{t}}(s) = -R^{\mathrm{t}}. \tag{23}$$

7) Minimising Energy Consumption: The rate of energy usage or the cycling power P_{cyc} produced by the cyclist is composed of four terms associated with the acceleration P_{ac}, tire rolling resistance P_{tr}, aerodynamic drag P_{dr}, and also the road slope P_{rd} [33]. Hence, if the effective headwind speed at time k is $v_{\mathrm{w}}(m/s)$, the simplified equation of output power required for cycling at speed v and having acceleration u takes the following form:

$$P_{\mathrm{cyc}}(v, a) = \underbrace{(m + m_{\mathrm{w}})a\, v}_{P_{\mathrm{ac}}} + \underbrace{C_{\mathrm{tr}} mgv}_{P_{\mathrm{tr}}} \\ + \underbrace{0.5\rho v\big(v + v_{\mathrm{w}}\big)^2 C_{\mathrm{dr}} A_{\mathrm{f}}}_{P_{\mathrm{dr}}} + \underbrace{mgve}_{P_{\mathrm{rd}}}, \tag{24}$$

where m (kg) is the mass of the bicycle and the rider, m_{w} (kg) is the effective rotational mass of the wheels and the tires, g (m/s^2) is the acceleration due to gravity, A_{f} (m^2) is the frontal surface, ρ (kg/m^3) is the density of the air, e is the slope of the road, and C_{dr} and C_{tr} are the coefficients of the aerodynamic drag, and rolling resistance respectively. To generate a speed trajectory which takes into account the energy consumption of the cyclist, in the reward function, we can use (24) to penalise energy consumption in each time interval as follows:

$$F^{\mathrm{e}}(s, a) = \Delta t \frac{P_{\mathrm{cyc}}(v, a)}{P^{\max}}, \tag{25}$$

where $P^{\max}$ is defined for normalisation as:

$$P^{\max} = (m + m_{\mathrm{w}})u^{\max} v^{\max} + C_{\mathrm{tr}} m g v^{max} \\ +0.5\rho v^{\max}\left(v^{\max} + v_{\mathrm{w}}^{\max}\right)^2 C_{\mathrm{dr}} F + m g v^{\max} e. \quad (26)$$

It should be noted that the power generation ability of a cyclist, or the cyclist's maximum possible speed or acceleration is assumed to be unchanged during his/her trip. Moreover, although there are research results suggesting that under certain circumstances, such as in cycling races, the fatigue of the cyclists plays a critical role [34], compared to cycling races, the role of fatigue for urban non-racing trips may be less. Having that said, the modular nature of the proposed framework allows for the inclusion of a dynamic fatigue model if necessary.

IV. Problem Formulation

Given the description of the process in Section II and the performance metric of interest detailed in Section III, we have all the required ingredients to formulate our problem concisely. Knowing the instantaneous state of the cyclist as well as the traffic light's state, our goal is to recommend optimal acceleration to the cyclist at each time step k that maximises the expected return, in terms of the personalised performance metric of that cyclist for his/her trip. To achieve that, we examine the following optimization problem for all $S_k \in \mathcal{S},\ k \in \{0, 1, \ldots, T\}$:

$$\begin{aligned} \max_{\pi}\ & \mathbb{E}\Big[\sum_{m=k}^{T-1} \gamma^{m-k}\ \overbrace{R\big(S_m, \pi(S_m), S_{m+1}\big)}^{r_{m+1}}\Big| S_k = s\Big] \\ s.t.\ & \text{Equation (13)} \\ & S_m \in \mathcal{S}, \qquad \forall m \in \{k, k+1, \ldots, T\} \\ & \pi(S_m) \in \mathcal{A}, \quad \forall m \in \{k, k+1, \ldots, T\} \end{aligned} \quad (27)$$

where T denotes the time step that the cyclist reaches the end point in Figure 1 and $\pi(s)$ is the policy to determine the control action in each state, i.e. $u(k) = \pi(S_k)$. Given the process state S_k, the optimization objective in (27) consists of the instantaneous performance metric r_{m+1}, discounted with the discount factor γ, accumulated over time, and averaged due to the stochastic transition of the process's states as expressed in (13)). In optimizing this objective, three constraints need to be respected: 1) the process dynamics described by (13) 2) the constraint on the admissible set of states described in (12), and 3) the constraint on the admissible set of control input. As the result of this optimization problem, we will achieve the optimal policy $\pi^*(s)$ that would assign the optimal acceleration at each time step k, given the state S_k i.e. $u^*(k) = \pi^*(S_k)$.

Since the state transition probabilities of the process are assumed to be known, Stochastic Dynamic Programming (SDP) is an appropriate method for solving the optimization problem in (27).

V. Stochastic Dynamic Programming

This section presents the stochastic dynamic programming approach used for solving the optimisation problem presented in Section IV. SDP is a powerful tool that has advantages over algorithms such as Monte Carlo simulation or Genetic Algorithm methods because it can give the exact optimal strategy for all possible state without using sampling or approximation. We will make use of the value iteration algorithm to compute the optimal policy $\pi^*(s)$. An essential prerequisite to do so, is to define $V^{\pi}(s)$ which quantify the value of state s under policy π. If each state transition at time step k is evaluated by the instantaneous reward r_{k+1}, then $V^{\pi}(s)$ represents how *good* it is, in terms of the expected cumulative reward, to be in state s and follow policy π thereafter until reaching to the terminal state at time step T:

$$V^{\pi}(s) = \mathbb{E}_{\pi}\Big[\sum_{l=0}^{l=\infty} \gamma^{\,l}\, r_{k+l+1} \mid S_k = s\Big]. \quad (28)$$

Note that without loss of generality in our problem, we allow $T = \infty$, assuming that when the system reaches the terminal point, it stays there forever and is not receiving any more reward. Moreover, the term γ is the discount factor through which the current value of the future reward can be altered. The discount factor ensures that the cumulative reward remains finite over infinite time horizon.

Compared to any other policy, the optimal policy $\pi^*(s)$ is the policy that if taken, results in the highest sate value (or expected rewards) among all possible policies. If $V^*(s)$ indicates the optimal value function of state s achieved by the optimal policy, the following Bellman optimality equation will be held that shows how the optimal value function of states in the state set $\mathcal{S}$ relates with each other,

$$V^*(s) = \max_{a\in\mathcal{A}} \sum_{s'\in\mathcal{S}} P_{ss'}^{a}\left(R_{ss'}^{a} + \gamma V^*(s')\right), \quad (29)$$

where

$$P_{ss'}^{a} = \mathbb{P}[S_{k+1} = s' \mid S_k = s,\ u(k) = a], \quad (30)$$

$$R_{ss'}^{a} = [r_{k+1} \mid S_k = s,\ S_{k+1} = s',\ u(k) = a]. \quad (31)$$

In general, no closed-form solution exists for solving the Bellman optimality equation, but various iterative algorithms including the value iteration algorithm described in Algorithm 1 can be used to find $V^*(s)$ and the optimal policy $\pi^*(s)$ in all the states [35].

Initialisation of $V(s)$ may vary for each problem. In this article, $V(s)$ is initialised to zero for all states. Moreover, the value function of the states that corresponds to the end point of the trip do not change.

VI. Simulation Case Studies and Results

Through three case studies, this section analyzes the properties of the proposed SDP approach by comparing its performance against a baseline acceleration policy as well as an existing approach in the literature. For all the case studies, we simulate a cyclist traveling in the direction of stream 02 (from East to West) towards the junction shown in Figure 2a. The total travel distance is assumed to be 290 meters and the intersection is located 250 meters downstream of the start point. The 40 meters downstream of the intersection will be enough for the cyclist to accelerate and cruise in his desired

Algorithm 1: Value Iteration Algorithm for Solving the Bellman Optimality Equation (29)

```
Initialize the value function V(s), ∀s ∈ S;
δ = 10^(−8), Δ = ∞;
while Δ > δ do
    Δ = 0;
    forall s ∈ S do
        V̄ ← V(s);
        V(s) ← max_{a∈A} Σ_{s'∈S} P^a_{ss'} (R^a_{ss'} + γ V(s'));
        Δ ← max(Δ, |V̄ − V(s)|)
    end
end
forall s ∈ S do
    π*(s) = argmax_{a∈A} Σ_{s'∈S} P^a_{ss'} (R^a_{ss'} + γ V(s'))
end
```

TABLE III
WEIGHTS USED IN THE THREE TYPES OF POLICIES

	W^f	W^i	W^c	W^d	W^s	W^t	W^e
NoStop-Prf-I	10^7	3	3	3	10	0	0
NoStop-Prf-II	10^7	3	3	10	10	0	0
Energy-Prf-I	10^7	3	3	3	0	0	10
Energy-Prf-II	10^7	3	3	10	0	0	10
Time-Prf-I	10^7	3	3	3	0	10	0
Time-Prf-II	10^7	3	3	10	0	10	0

speed before reaching the end point. This allows for a fair comparison of energy and travel time between trajectories where the cyclist has to or doesn't have to stop at the traffic signal.The state of the traffic light, denoted as s_l and defined in (11), is assumed to follow the transition probabilities described in Tables VII-X in Appendix B. In general, these transition probability can be obtained from historical data.

In the case studies, we focus on three types of policies, namely NoStop-Prf, Time-Prf, and Energy-Prf, where each policy is associated with a cycling preference: minimizing the chance of stop during the trip, the travel time, and the energy consumption, respectively. Each policy has been generated by deploying the numerical scheme described in Section V and are distinguished from each other due to the use of different set of weights in (14). The weights chosen for generating each type of policy are given in Table III. Note that two variations are used for each policy type to account for the level of the cyclist's willingness to keep his desired speed. The lower value of W^d associates with less resistance from the cyclist in deviating from his desired speed v^d.

In the remainder of this section, we first describe the baseline model with which we benchmark the performance of the three aforementioned policies. The baseline model is used to describe the cyclist's manoeuvre towards an intersection with no speed advice to follow. Next, in three case studies, we analyse the performance of the three SDP-generated policies and compare it with the policy generated from the baseline model, with each other, and also with the most relevant approach available in the literature.

A. Baseline Model

If cyclist is within human-vision distance D_h from a traffic light, we assume that the traffic signal status will have an effect in the cycling behaviour. However, no model could be found in the literature that describes and calibrates the deceleration and acceleration of cyclists in response to the colour of the traffic light. That being the case, to have a baseline model for comparison with the proposed control approach, the following description is used for cyclists. As explained in Appendix A, if $d_j(k)$ denotes the distance between the cyclist and traffic light at time step k, and $C_s(k)$ denotes the number of required time steps in order to fully stop before the intersection, we allow the interaction of a cyclist with the traffic light by defining the cyclist's acceleration/deceleration in the baseline model as follows:

$$\bar{u}(k) = \begin{cases} -\dfrac{v(k)}{C_s(k)\Delta t}, & \text{If Case 1} \quad (32) \\ 0, & \text{If Case 2} \quad (33) \\ u'\left(1 - \left(\dfrac{v(k)}{v^\mathrm{d}}\right)^2\right), & \text{Otherwise} \quad (34) \end{cases}$$

where

$$C_s(k) = \max\left(1, \lfloor \frac{2d_j(k)}{v(k)\Delta t} \rfloor\right), \tag{35}$$

u' is the comfortable acceleration of the cyclist, and

$$\text{Case 1:} 0 < x^\mathrm{s} - x(k) < D_h, B \in B^s, \tag{36}$$

$$\text{Case 2:} 0 < x^\mathrm{s} - x(k) < D_h, B \notin B^s, v > v^\mathrm{d}. \tag{37}$$

According to (32)-(34), in the human-vision zone, while the light is red, the cyclist decelerates according to (32) until he eventually stops. Note that decelerating according to (32) prevents negative speed and guarantees full stop of the cyclist at or upstream of the traffic light. If the light gets green while the cyclist is in the human vision distance to the intersection, the cyclist keeps his current speed, unless it is lower than his comfortable speed. In the latter case, he accelerates according to (34) to reach the comfortable speed v^d.

In the following sections, this baseline acceleration policy, hereinafter referred to as NoControl policy (NC), will be used for benchmarking against the policies generated by the SDP approach. The simulation parameters used to generate the experiments are given in Table IV. Moreover, the speed, distance and acceleration are discretised with the steps of 0.25, 0.5, and 0.25 respectively.

B. Case Study I - The Monte Carlo Simulations

The aim of this case study is threefold: 1) to benchmark the performance of the SDP-generated policies against the NoControl policy, by investigating their average results in randomly generated runs of experiments, 2) to examine the

TABLE IV
SIMULATION PARAMETERS

$x^{\max}$(m)	$v^{\max}(m/s)$	$u^{\min}$(m/s^2)	$u^{\max}$(m/s^2)	Δt(s)	x^{s}(m)	κ(m/s)	v^{i}(m/s)	R^{f}	R^{s}
290	7.75	-1.5	0.75	2	250	0.4	1	1	1
R^{t}	m(kg)	m_{w}(kg)	C_{tr}	ρ(kg/m^3)	v_{w}(m/s^2)	C_{dr}	e	A_{f}(m^2)	D_{h}(m)
1	95	0.95	.008	1.226	0	1.2	0	0.616	30

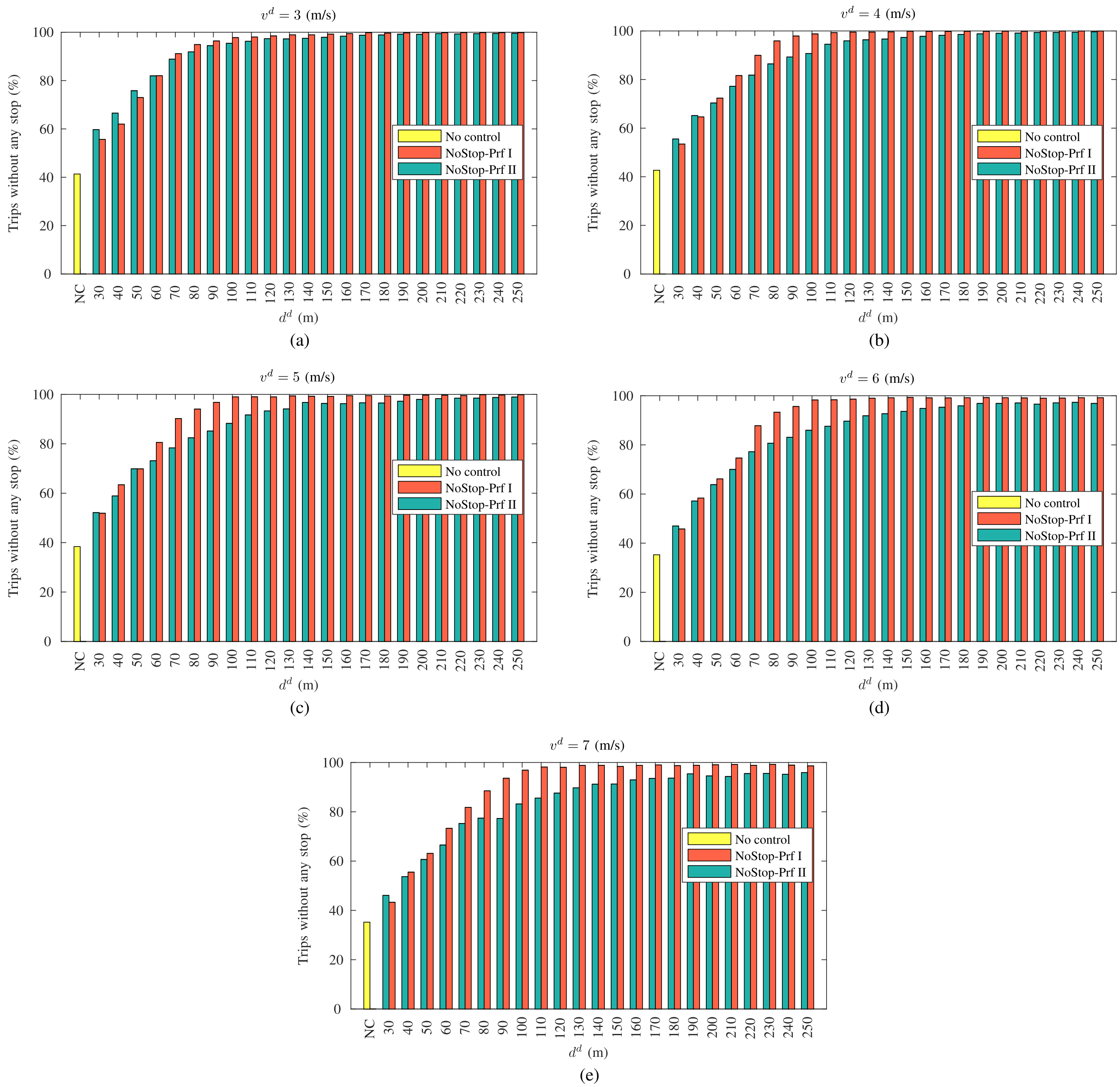


Fig. 4. Percentage of case where the cyclist can pass the light without stopping for the NoControl policy (yellow bar) and NoStop-Prf policies for variation of v^{d}. The red bar associated with NoStop-Prf I and the green bar associated with NoStop-Prf II policy. The rate in NoStop-Pref policy vary depending on from where the optimal advice is given (d^d).

impact of the cyclists' desired speed on the performance, and 3) to analyze the distance over which a speed advice improves the performance.

To achieve these goals, in the first step, for each desired speed $v^{\mathrm{d}} \in \{3, \ldots, 7\}$ (m/s), all six variations of policies (NoStop-Prf-I&II, Time-Prf-I&II, and Energy-Prf-I&II) are generated. In other words, for each v^{d}, we have six different policies corresponding to three different cycling preferences, in accordance with Table III. Then, in order to address the third goal of the case study, we vary the location that we start giving

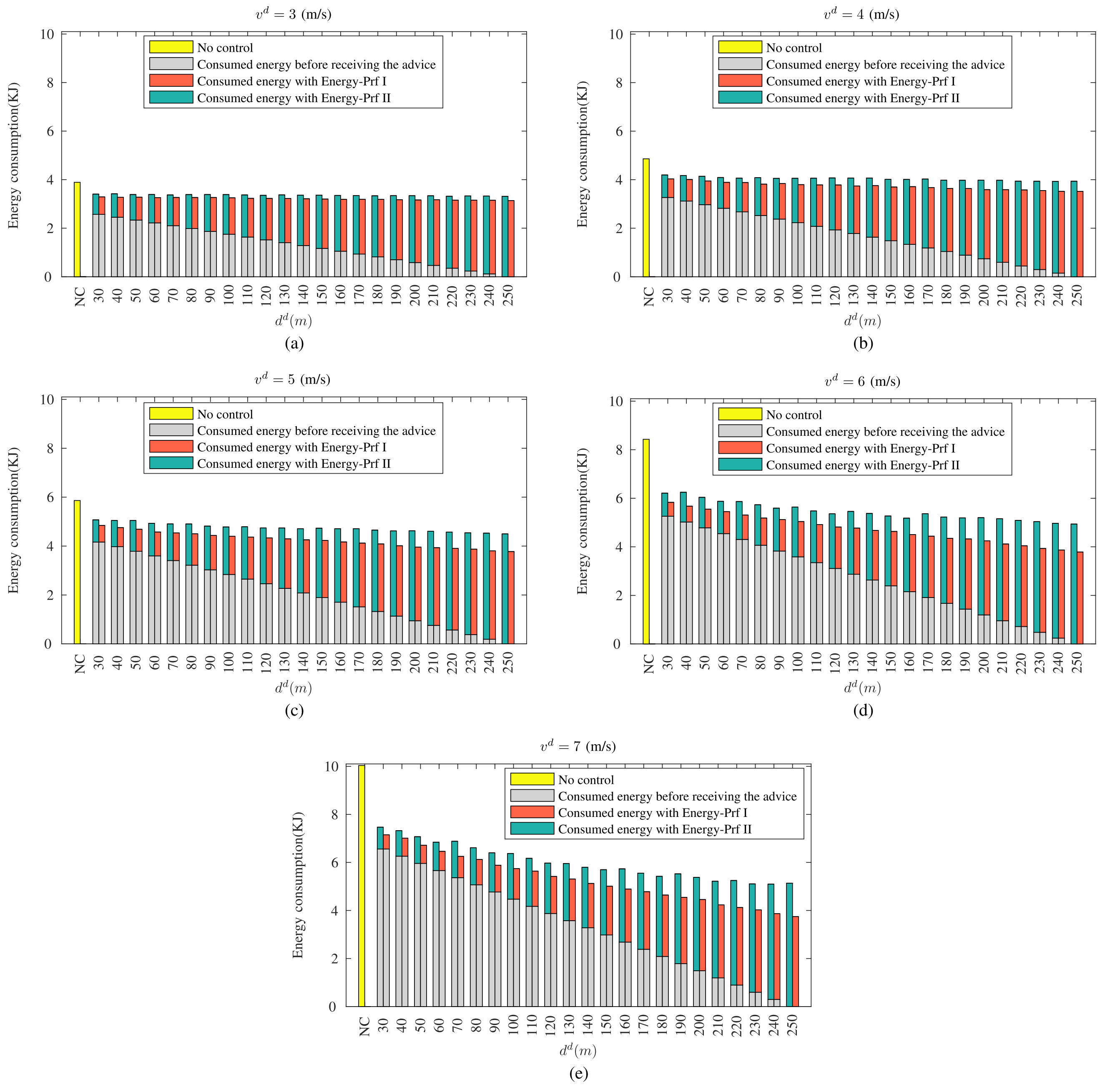


Fig. 5. Energy consumption cost for NoControl policy (yellow bar) and Energy-Prf policies for variation of v^{d}. The red bar and the green bar associated with the energy consumed from the point that the cyclist gets the advice and follow Energy-Prf I and Energy-Prf II policy respectively. The grey bar indicates the amount of energy consumed by the cyclist from the start point until he receives the advice. The cost of Energy-Prf policy vary depending on from where the optimal advice is given (d^d).

the advice to the cyclist. To be more specific, for each $d^{\mathrm{d}} \in \{30, 40, \ldots, 250\}$ meters, we let the cyclist cycle at his desired speed and only starts to follow each of these policies if he is at d_d meters far from the intersection. With 6 alternative forms in policy, 5 variations in the desired speed v^{d}, and 23 variations in the start location of giving advice, we will have 690 different experiments. We run each of these experiments 10000 times where in each run, the initial state of the traffic light when the cyclist start to receive the advice is randomly generated in accordance with the transition probabilities. The average result of these experiments is depicted in Figures 4-6. Let us analyse the performance of the three types of SDP-generated policies and the impact factor of v^{d} and d^{d} on the performance improvement in each of these policies separately:

1) NoStop-Prf Policies: It can be seen from Figure 4 that by following NoStop-Prf-I policy, with 98% chance and regardless of his favourite speed, the cyclist who starts to get the optimal advice in $d^{\mathrm{d}} > 110$ can reach to the final point without any stop. That is to say, if the aim is to travel with no stop, the performance of NoStop-Prf-I policy does not improve if the cyclist starts to receive the optimal advice in distance far more than 110 meters from the intersection.

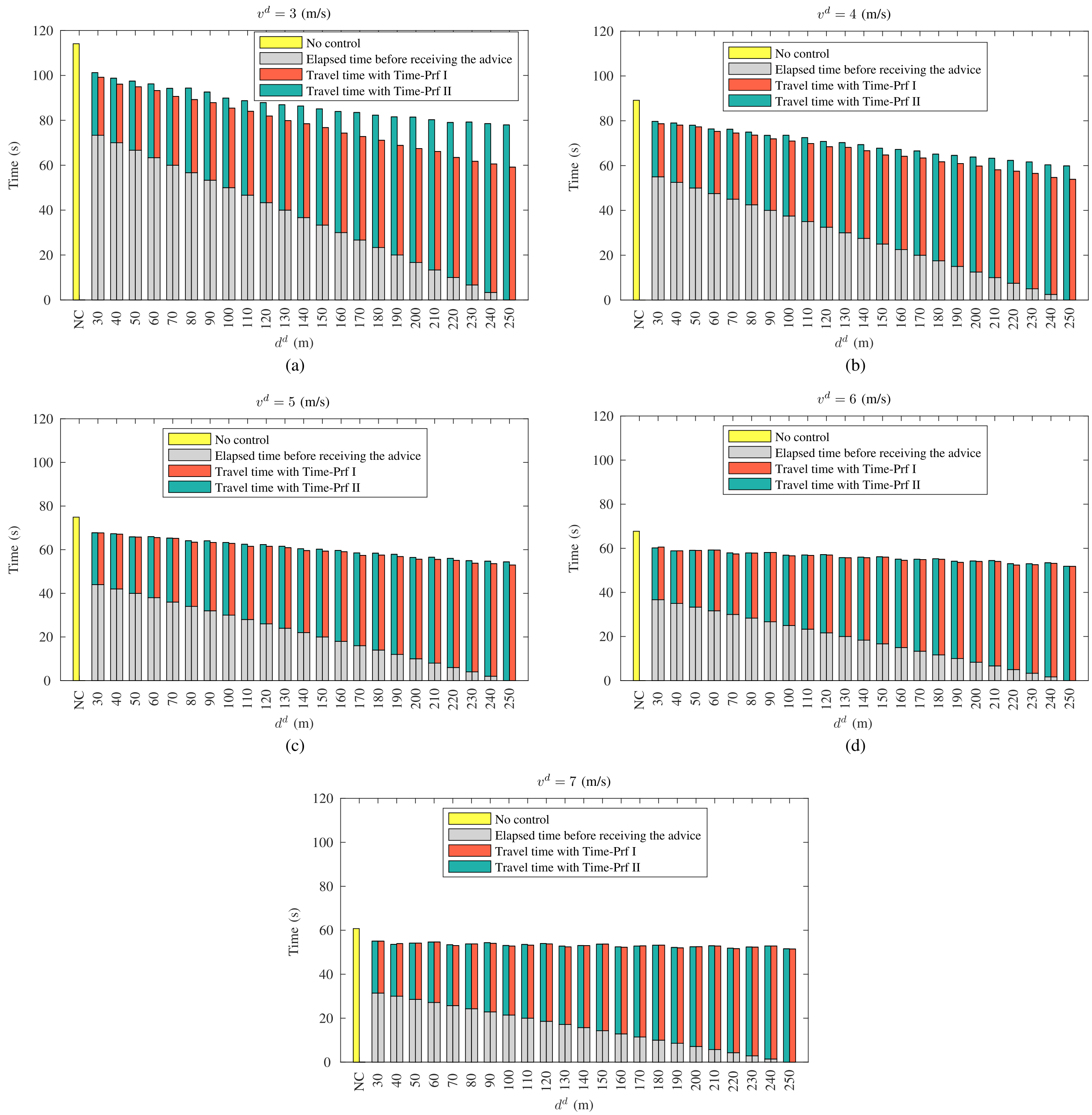


Fig. 6. Travel time for NoControl policy (yellow bar) and Time-Prf policy for variation of v^{d}. The red bar and the green bar associated with the travel time from the point that the cyclist gets the advice and follow Time-Prf I and Time-Prf II policy respectively. The grey bar indicates the time that takes the cyclist before he receives the advice. The travel time in Time-Prf policy varies with d^d.

The figure also shows that in general, NoStop-Prf-II underperforms NoStop-Prf-I policy as the advice of the former is for the cyclist who has more resistance in deviating from his desired speed. Though the performance difference between the two policies is minor if the advice is given in very short distance ($d^{\mathrm{d}} < 60$) or in long distance ($d^{\mathrm{d}} > 160$). In the former, neither of the policies has enough time to properly control the cyclist before reaching the intersection and in the latter, there is enough time for both policies for efficient control.

Moreover, for $d^{\mathrm{d}} < 100$, we observe that the chance of catching green without any stop gets higher if the favourite speed of the cyclist, which is the cyclist's speed before receiving any speed advice, is lower. The rationale behind this observation relates to the fact that with similar deceleration, a cyclist with lower initial speed can travel a fixed distance in longer time. Hence, if the light is red and the cyclist starts to receive the advice of deceleration, it is more likely that the light gets green before the cyclist arrives at the intersection, if his initial speed is lower. Simply because it can take him

TABLE V
PERFORMANCE EVALUATION OF ALL THREE POLICY TYPES IN CASE STUDY I

	Chance of no stop			Energy(kJ)			Time(s)		
	NoStop-Prf-	No control	Improvement	Energy-Prf	No control	Improvement	Time-Prf-	No control	Improvement
$v^{\mathrm{d}} = 3$	I:99.87% II:99.5%	41.34%	141.58% 140.83%	I:3.14 II:3.31	3.90	19.37% 14.81%	I:59.15 II:77.95	114.11	48.16% 31.68%
$v^{\mathrm{d}} = 4$	I:99.99% II:99.67%	42.68%	134.27% 133.52%	I:3.52 II:3.93	4.86	27.59% 19.15%	I:53.88 II:59.87	89.14	39.55% 32.83%
$v^{\mathrm{d}} = 5$	I:99.82% II:98.93%	38.40%	159.94% 157.63%	I:3.78 II:4.50	5.86	35.59% 23.28%	I:52.99 II:54.42	74.91	29.25% 27.35%
$v^{\mathrm{d}} = 6$	I:99.31% II:97.34%	35.28%	181.49% 175.90%	I:3.79 II:4.94	8.42	55.06% 41.39%	I:51.79 II:51.88	67.72	23.5% 23.39%
$v^{\mathrm{d}} = 7$	I:99.25% II:95.90%	35.22%	181.80% 172.28%	I:3.75 II:5.10	10.03	62.69% 49.22%	I:51.46 II:51.58	60.75	15.28% 15.08%

more time before reaching the intersection and the chance that the traffic light changes its colour gets higher. On this account, the lower the speed, the higher the success rate and hence, both policies, especially NoStop-Prf-II policy which has a higher weight on cruising at the desired speed, serves a cyclist better for $d^{\mathrm{d}} < 100$ if he has a lower desired speed.

From the figure, it can also be inferred that for all cases, there is a distance above which the advice does not improve the performance.

2) Energy-Prf Policies: Both of the Energy-Prf policies outperform the NoControl policy for all values of v^{d} in Figure 5. The advantage of giving the optimal advice raises when the cyclist's favourite speed is higher and his resistance to changing his speed is lower. This combination gives more room for more energy-efficient travel, i.e., reducing the speed for saving energy. From Figure 5 it is also notable that for $v^{\mathrm{d}} <= 5$, the energy that the cyclist may save hardly changes with d^{d}. That is to say, for $v^{\mathrm{d}} <= 5$, it is almost equally good if the cyclist starts to receive the advice only when he as close as 30 meters to the intersection. Quite the contrary, for $v^{\mathrm{d}} > 5$, the sooner the advice is given, the more benefit the cyclist may yield, in terms of energy, by following the advice. Note that although for many cases, the optimal advice derived from Energy-Prf policies slows down the cyclist, this does not hold invariably. In other word, an energy-efficient manoeuvre is not limited to deceleration. At some situations, the cyclist might be advised to speed up, e.g. to catch the green light, in order to prevent consuming unnecessary energy in the future by stoping in front of the red light and use more energy to reach to his desired speed again. Similar to the previous case, there is a distance above which the advice does not enhance the performance.

3) Time-Prf Policies: As seen in Figure 6, the performance of the Time-Prf policies surpass that of the No Control policy for all values of v^{d}. Moreover, the benefit of following the Time-Prf policy type is more notable if the cyclist's favourite speed as well as his resistance to deviate from his favourite speed is lower. Since the upper bound of the cyclist's speed is set to 7.75, compared to the the cyclist with higher v^{d}, the one with the lower v^{d} has more room to increase his speed before reaching the upper bound. Comparatively, the benefit that the latter can receive from the Time-Prf policies is higher. From Figure 6, it is also clear that for $v^{\mathrm{d}} >= 6$, an increase in d^{d} does not make a significant change in the travel time, i.e., there is not much gain in the performance if we start to give the advice in far distance from the intersection. On the contrary, for lower desired speed, the earlier the advice is started to be given, the lower the total travel time. It is important to emphasise that although acceleration is a natural way for saving time, the optimal advice derived from Time-Prf is not restricted to that only. As demonstrated in the next case study, in some situations and when the light is red, the cyclist is advised to first decelerate or even stop at some distance upstream of the intersection, in order to start accelerating at the right time and catch the green light having high speed. The figure also shows that there is a maximum distance beyond which there is no more improvement due to the advice.

Table (V) summarizes the results of Case study I. In this table, for each value of v^{d}, we have compared the performance of No Control policy with the best performance of the SDP-generated policies that can be achieved among all values of d^{d}. We have also outlined the maximum enhancement yielded by deploying the SDP-generated policies. As it is observed from the table, the chance for traveling without stop may raises up to 181.8 % if NoStop-Prf policy type is followed by the cyclist. Energy-Prf policy type can help the cyclist to save up to 62.69% energy and Time-Prf policy type may reduce the travel time of a cyclist by 48.16%.

C. Case Study II - Personalisation in the SDP-Generated Policies

One of the important features of the SDP-generated policies is that they can be personalised. Knowing the personal preferences, the advice can be tailored to meet the need of individual

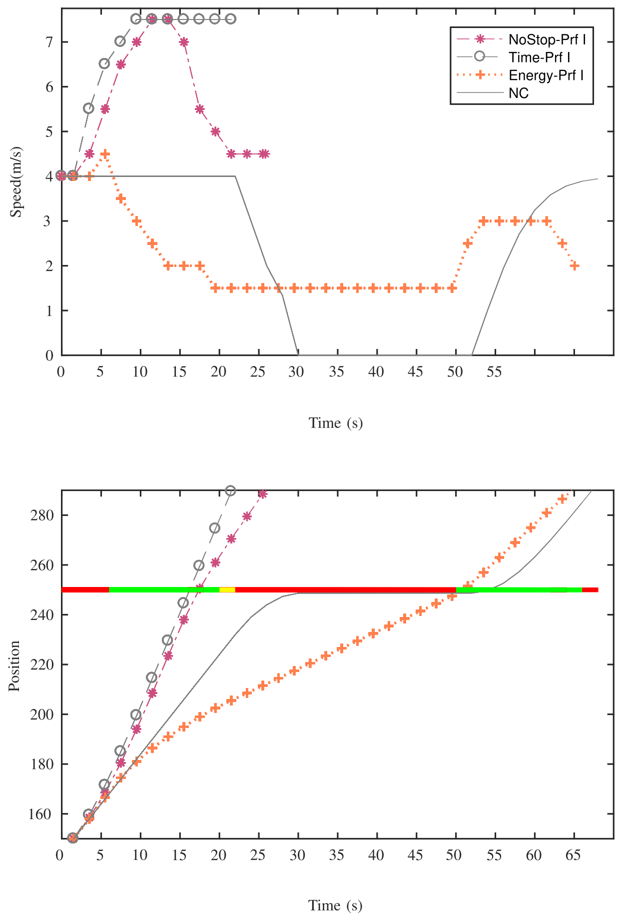


Fig. 7. Position and speed of the cyclist in case study II-A with the NoStop-Prf I, Time-Prf I, Energy-Prf I, and NC policies.

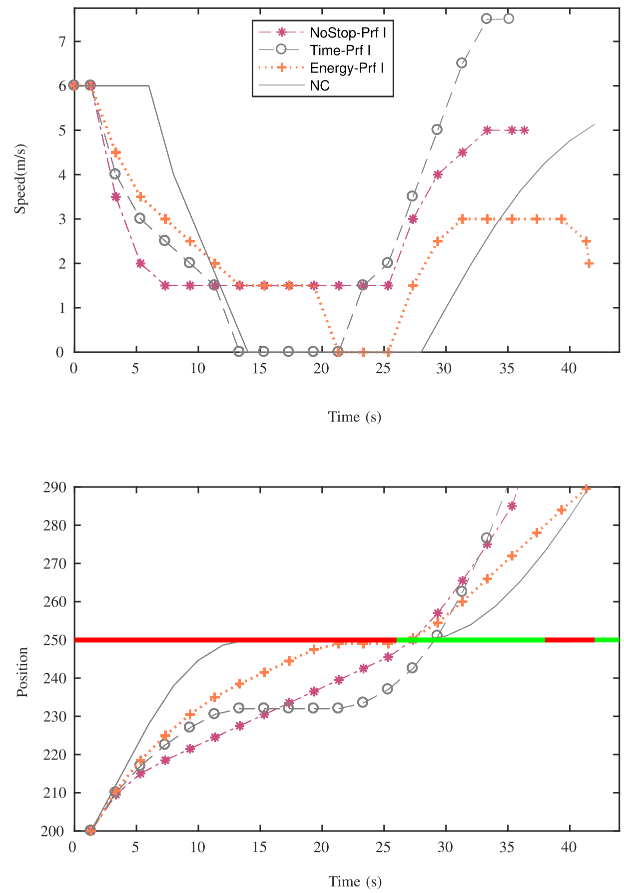


Fig. 8. Position and speed of the cyclist in case study II-B with the NoStop-Prf I, Time-Prf I, Energy-Prf I, and NC policies.

cyclists. That is to say, in similar traffic light conditions, each policy may give an advice that is different from that of other policies. The difference between the optimal advice generated from each policy under similar traffic condition is illustrated by the following two examples called Case study II-A and Case study II-B.

The results of Case study II-A, depicted in Figure 7, demonstrates the different resulting trajectories for a cyclist when v^{d} and d^d is chosen as 4 (m/s) and 100 (m) repectively. The advice is generated for three different policies namely Time-Prf I, NoStop-Prf I, and Energy-Prf I, and is compared with the NC policy. The proposed algorithm can predict that, in the worst case that the upcoming green light does not extend, there will be enough time for the cyclist to reach the end of green if he accelerates to the highest possible speed. This is what the cyclist is advised to do under Time-Prf I and NoStop-Prf I policies. As seen in the figure, such manoeuvre helps the cyclist to pass the first green light, that is the optimal decision in terms of the two policies' performance metric; in the case of Time-Prf I, speeding up obviously leads to shorter travel time, and in the NoStop-Prf I case, speeding up leads to passing the junction with lower total cost than the slowing down for red and aiming for catching the second green.

After passing the junction, the advice in the two policies differ as one suggests to keep the high speed to minimise the travel time and the other one suggests to decelerate and smoothly minimize its deviation from the desired speed. In comparison, in Energy-Prf case, which energy consumption comes to matter, the algorithm advises to cycle at relatively low speed and smoothly move toward the junction and catch the second green light. Clearly, the baseline algorithm does not help the cyclist in catching the green and the cyclist needs to stop at the red light and use a lot of energy to accelerate from zero speed after the light gets green. Table VI clearly shows that cyclist following the advice from Energy-Prf I policy uses significantly less energy to cycle the path.

In Case study II-B, we simulate the trajectories resulting from the four policies in a different traffic light setup shown in Figure 8. This time, the traffic light just got red as the cyclist starts to receive the advice in 50 meters upstream the intersection. It is interesting to see from the figure that the algorithm suggests the cyclist to have a full stop in distance and start to accelerate after few seconds of standstill in order to minimise the total travel time in the case of Time-Prf I. With No-Stop-Prf I, the cyclist needs to diverge from the desired speed as much as necessary to guarantee catching the beginning of green. The movement in Energy-Prf is smooth and use the minimum amount of energy thought results in a stop at the red light. As clearly seen in Table VI, the amount of energy is the lowest among all if Energy-Prf I policy is followed.

TABLE VI
PERFORMANCE OF THE FOUR POLICIES IN CASE STUDY II. Y:YES, N:NO

	Stop		Energy(kJ)		Time(s)	
	Case study II-A	Case study II-B	Case study II-A	Case study II-B	Case study II-A	Case study II-B
NC	Y	Y	2.7	2.8	68.0	42.0
NoStop-Prf-I	N	N	4.1	1.9	25.8	36.0
Time-Prf-I	N	Y	6.3	3.4	21.5	35.1
Energy-Prf-I	N	Y	1.5	0.7	65.1	41.0

D. Case Study III- Comparison With the Available Approach in the Literature

In the third case study, we compare the algorithm proposed in this article with the one suggested in [16]. As it is already mentioned in the introduction, the algorithm developed in [16], hereafter we call Prob-SPAT as suggested in [16], is an alternative algorithm in the literature that takes into account the probabilistic prediction of traffic signal timing. Prob-SPAT uses speed (v) and time (t) as states to describe the kinematics of vehicle traveling towards an intersection while x is an independent variable. With such setting, if $\Delta x = x_{i+1} - x_i$ where x_i and x_{i+1} indicate consecutive discrete locations, is the sampling interval, the vehicle kinematics can be described as:

$$v_{i+1} = \sqrt{v_i^2 + 2a_i \Delta x} \tag{38}$$

$$t_{i+1} = t_i + \frac{2\Delta x}{v_i + \sqrt{v_i^2 + 2a_i \Delta x}}, \tag{39}$$

The control variable is acceleration a_i, which is given to the vehicle in each M meters. The advice aims to minimise the following cost function, indexed over position x with index i:

$$J = \sum_i w_1' \frac{t_{i+1} - t_i}{\Delta t_{min}} + w_2' |\frac{a_i}{a_{max}}| + w_3' c(x_i, t_i) \log_e(\mathrm{pr}(x_i, t_i)) \tag{40}$$

where Δt_{min} is the minimum time to complete the step if starting and ending at the maximum velocity and is used as a normalizing factor, a_i is the constant acceleration assumed during step i, and a_{max} is the maximum allowed acceleration. The constants w_1', w_2' and w_3' are weighting terms. Motion constraints imposed by the traffic signal's red time interval, are imposed as a soft constraint by inclusion of the term $c(x_i, t_i)\log(\mathrm{pr}(x_i, t_i))$ in the cost function. The value of $c(x_i, t_i) = 1$ if there is a traffic light at position x_i, otherwise it is zero. Moreover, $\mathrm{pr}(x_i, t_i)$ represents the probability of green at time t_i for a light situated at position x_i. Having the current colour of the light located at position x_i, [16] suggests a method to find the probability of green light in t_p time step ahead. The optimisation problem is solved using *Deterministic* Dynamic Programming but in a receding horizon manner. Interested readers are referred to [16] for more details. Having (38)-(40), let us outline the main differences of the SDP-generated approach suggested in this article with the Prob-SPAT approach in [16]:

- Unlike in Prob-SPAT, which needs the cycle time of the traffic signal to be fixed, SDP-generated approach does not make any assumption on the type of traffic signal timing and is suitable to be used efficiently in, e.g., fixed-time and traffic-responsive junction signalisation alike.
- The probabilistic prediction of traffic light timing in SDP-generated approach is not only based on the current colour of the traffic light but also on how long the traffic light have had this colour. The approach also takes into account how the traffic light evolves in time. If this information is available, the resulting advice from SDP-generated approach may overperform that of Prob-SPAT approach.
- SDP-generated approach pays special attention to the personal preferences of the user and in that respect, is significantly richer than Prob-SPAT.
- According to [16], each time the new information about the status of traffic light is available, a new deterministic dynamic programming is needed to be solved over the remaining trip horizon. On contrary, in the suggested approach presented in this article, the stochastic dynamic programming is solved off-line and only once. Hence, the online computational burden is much lower and there is much more flexibility in choosing a finer discretisation in the SDP-generated approach. It is worth to mention that for using Prob-SPAT algorithm, it is also possible to generate a table of all possible state combinations and the corresponding optimal action offline, though this is not the approach taken in [16].
- According to [16], a low-level controller verifies and override the velocity recommendation if the planner makes a recommendation that would pass through a red light. Such low-level controller is not required in the SDP-generated approach, since as explained in Section III, any action that leads to passing a red light is explicitly and significantly penalised.
- According to (38), in the Prob-SPAT approach, zero acceleration is not a feasible action for a cyclist with zero speed, i. e., if the cyclists stops, he will remain standstill for ever. As it is demonstrated in the previous case study, in some situations, a full stop can be the optimal action in the SDP-generated approach.

Since in (40), the travel time is included in the cost function, among all the SDP-generated policies, Time-Prf type policy

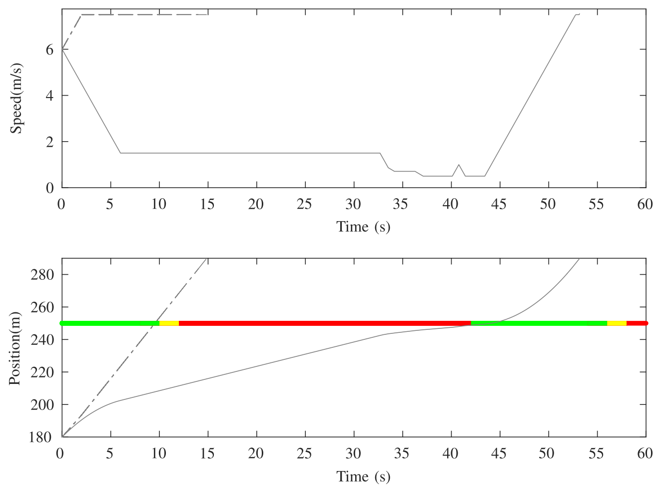


Fig. 9. Comparison of the Prob-SPAT (solid line) and Time-pref (dashed dotted line) algorithm.

is chosen as the most suitable one to be compared with Prob-SPAT. To deploy Prob-SPAT, the duration of green and red lights are respectively set to 14 and 30 seconds, as the result of averaging over 1000 random runs. The value of w_1', w_2' and w_3' in (40) are chosen as $1/8$, 0 and 100 respectively and time and speed space are discretised with the step of 1 and 0.25 respectively. Moreover, M in Prob-SPAT is chosen to be 0.5 meters, meaning that a new control input must be calculated in each 0.5 meters.[1]

The simulation result is depicted in Figure 9 and clearly shows the difference between the two methodologies through this experiment. Since in Time-Prf I, the algorithm is aware of how long the light has been green and how the state of the traffic light will evolve in time, it advises the cyclist to accelerate from the beginning and catch the first green light. The Prob-SPAT algorithm is not equipped with such information. The resulting advice is not helpful to catch the first green light though is successful in advising to catch the second green light without any stop.

VII. Further Discussion

This section includes discussions about the practical implementation of the algorithm and future direction of this research.

A. Real-Time Implementation

The result of the value iteration algorithm is a policy map, which is produced offline and can be deployed with low computation burden in real time. The policy map relates the states of the system, defined as in (12), to the control input (acceleration). Hence, the only on-board computational requirement is interpolation between the map's grid point. By choosing the size of the grid, we can make a trade-off between the memory storage and the performance of the algorithm.

[1]The chosen value of M in [16] is 20 meters. However, to have a fair comparison, for Prob-SPAT, we have used the same discretization step size as the one used in the SDP-generated approach. Note that such fine discretised grid significantly increases the on-line computational time of Prob-SPAT and makes Prob-SPAT impractical to be used in a receding horizon setting.

B. The Choice of Discretisation Step Size

It is important to note that special care is required in choosing the discretization step size in time/location/velocity such that zero speed can be generated in (1). With the selected discretisation step size in the presented case studies, the off-line computation time of each policy on a Macbook with 1,2 GHz Intel Core M processor is 310 minutes on average. Finer discretization step size and acceleration may improve the efficiency but will increase the required off-line computation time and off/on-line memory as well.

C. Extension of the MDP Model

The MDP model used in this article can be extended to include other traffic-related events such as detector loop actuation, priority lane events or push buttons. Such extension may increase the accuracy of the model in describing traffic signal timing in real world. Moreover, by defining various sets of transition probabilities of signal timing in various time of the day, the advice given to the cyclist can be adjusted accordingly to be suited for, e.g., off-peak or morning/evening peak hours.

D. The Choice of Actuator

It is still an open question that what form of actuation is the most suitable way for communicating the advice to a cyclist. The form of communication should be telling enough, but at the same time should not cause a distraction for the cyclist. Visual feedback on the road, auditive feedback using smart phones, or haptic pushback force on the pedals, are among the potential ways for communicating the personalised advice with cyclists. More studies are required to investigate the level of comfort of cyclists in response to each of these designs.

E. Alternative Form of Advice

The action set selected in this note is the set of feasible acceleration for an average cyclist. Depending on the type of actuation, the type of action may change as well. For instance, for humans, probably speed advice is more intuitive, but for e-bikes, the acceleration power can be more directly connected to the electro-mechanical system. Even if the action is formulated differently, similar approach can be used to find the optimal policy and no significant changes are expected in the results.

F. Compliance of the Cyclist

It is assumed in this work that the cyclist is able to fully comply with the advice. In [36], the authors have addressed the case where the cyclist may not be able, or do not wish to follow the given advice fully. In such case, a reinforcement learning-based architecture that combines learning and planning is proposed that learns the cyclist's behaviour in relation to the advice and plans the best next move of the cyclist on-the-fly.

TABLE VII
TRANSITION PROBABILITY $p(s_l'|s_l)$ WHEN $B_b = B_1$

$p(s_l'|s_l)$ — s_l' / s_l	$(B_1,\ n_1^{1+},\ n_1^{2+})$	$(B_6,\ n_1^{1+},\ 1)$	$(B_8,\ 1,\ n_1^{1+})$	$(B_{12},\ 1,\ 1)$
$(B_1,\ 0 < n_1^1 < n^{min},\ 0 < n_1^2 < n^{min})$	1	0	0	0
$(B_1,\ n^{min} \leq n_1^1 \leq n_+^{min},\ n^{min} \leq n_1^2 \leq n_+^{min})$	0.5	0.3	0.2	0
$(B_1,\ n_+^{min} < n_1^1 \leq n^{max},\ n_+^{min} < n_1^2 \leq n^{max})$	0.1	0.6	0.3	0
$(B_1,\ n_1^1 = n^{max},\ n_1^2 = n^{max})$	0	0	0	1

G. Speed Advisory System for a Group of Cyclists

This work can be seen as an analogy to the studies in the literature on driver assistance systems for a single vehicle, i.e., it is assumed that the movement of a cyclist is not influenced by other cyclists. While the literature on the design of speed advisory systems for a group of vehicles is rich, the topic is still novel for bicycles [37] and has not yet been addressed for the case in signalised urban areas. It requires accommodating bicycle following model in the problem formulation and it is deemed an important research direction for further study.

VIII. CONCLUSION

This article puts forward a new approach that takes into account the stochastic nature of traffic light phasing and timing and gives optimal acceleration advice to a cyclist in regard to his preferences in cycling. The process is formulated as a Markov Decision Process with appropriate action set and reward functions which reflect the personal preferences of a typical cyclist in an urban area. These preferences include minimising travel time, minimising energy consumption, or minimising the chance of stop at the red light. Stochastic dynamic programming or more precisely, the value iteration algorithm is used for controlling this MDP. The results of different control decisions based on various cycling preferences has been compared in multiple case studies. Depending on the chosen control objective, it shows an improvement of up to 181% in the probability of finishing the trip without any stop, up to 62% saving in the energy and up to 48% saving in the travel time. Moreover, for all three control objectives it holds that there is a distance beyond which the performance doesn't improve anymore. Meaning that outside this region, the cyclist doesn't have to be controlled, and it is not needed to determine the acceleration advice. Furthermore, this distance depends on the objective, and the cyclist's desired speed. In addition, the different objectives also result in very different cyclist trajectories, and thus personalization makes sense. The proposed algorithm may be combined in the future with speed advisory systems for motorised vehicles in order to find a trade-off for serving users of various mode of transportation in urban areas.

APPENDIX A
BASELINE MODEL

We are interested to find the required deceleration that a cyclist should take such that he stops after traveling at most d_j meters. We are also interested to know the time that takes the cyclist to stop. For that, let us start with (1). If v_0 and u denote the initial speed and constant acceleration of a cyclist respectively, from (1), it is easy to show that the cyclist's speed after n time steps will be obtained as

$$v(n\Delta t) = v_0 + un\Delta t. \tag{41}$$

Hence, if the cyclist constantly decelerates with $u < 0$, it takes him n time steps before he fully stops, where:

$$u = \frac{-v_0}{n\Delta t} \tag{42}$$

Note that n is a positive integer. Having (1) and (42), with constant deceleration of u, the distance to travel before a full stop should respect:

$$d_j = \frac{-v_0^2}{2u} \tag{43}$$

Hence, u must satisfy:

$$u = \frac{-v_0^2}{2d_j}. \tag{44}$$

Comparing (44) with (42) and knowing $n \in \mathbf{Z}^+$, let us choose

$$n = C_s = \max\Big(1, \lfloor\frac{2d_j}{v_0\Delta t}\rfloor\Big), \tag{45}$$

Replacing (45) in (42), the required deceleration for a full stop after C_s time step can be obtained as:

$$\tilde{u} = \frac{-v_0}{C_s\Delta t}. \tag{46}$$

Having such deceleration, the real traveling distance before a full stop will be $\tilde{d}_j < d_j$ where:

$$\tilde{d}_j - d_j = \frac{v_0\Delta_t}{2}(C_s - n). \tag{47}$$

TABLE VIII

TRANSITION PROBABILITY $p(s_l'|s_l)$ WHEN $B_b = B_2$

$p(s_l'\|s_l)$ s_l' s_l	$(B_2,\ n_2^{1+},\ n_2^{2+})$	$(B_7,\ 1,\ n_2^{2+})$
$(B_2,\ 0 < n_2^1 < n^{max},\ n_2^2)$	0.4	0.6
$(B_2,\ n_2^1 = n^{max},\ n_2^2)$	0	1

TABLE IX

TRANSITION PROBABILITY $p(s_l'|s_l)$ WHEN $B_b = B_4$

$p(s_l'\|s_l)$ s_l' s_l	$(B_4,\ n_4^{1+},\ n_4^{2+})$	$(B_{10},\ 1,\ 1)$
$(B_4,\ 0 < n_4^1 < n^{min},\ n_4^2 < n^{max})$	1	0
$(B_4,\ n_4^1 < n^{max},\ 0 < n_4^2 < n^{min})$	1	0
$(B_4,\ n^{min} \leq n_4^1 < n^{max},\ n^{min} \leq n_4^2 < n^{max})$	0.3	0.7
$(B_4,\ n_4^1 = n^{max},\ n_4^2)$	0	1
$(B_4,\ n_4^1,\ n_4^2 = n^{max})$	0	1

TABLE X

TRANSITION PROBABILITY $p(s_l'|s_l)$ WHEN $B_b = B_5$

$p(s_l'\|s_l)$ s_l' s_l	$(B_5, n_5^{1+}, n_5^{2+})$	$(B_{11},\ 1,\ 1)$
$(B_5,\ 0 < n_5^1 < n^{min},\ 0 < n_5^2 < n^{min})$	1	0
$(B_5,\ n^{min} \leq n_5^1 \leq n_+^{min},\ n^{min} \leq n_5^2 \leq n_+^{min})$	0.8	0.2
$(B_5,\ n_+^{min} < n_5^1 \leq n^{max},\ n_+^{min} < n_5^2 \leq n^{max})$	0.2	0.8
$(B_5,\ n_5^1 = n^{max},\ n_5^2 = n^{max})$	0	1

APPENDIX B
TRANSITION PROBABILITIES

Signal phase and timing of the intersection in the case studies are assumed to evolve based on the transition probabilities

TABLE XI

TRANSITION PROBABILITY $p(s_l'|s_l)$ WHEN $B_b = B_2$

$p(s_l'\|s_l)$ s_l' s_l	$(B_2,\ n_6^{1+},\ 1)$	$(B_7,\ 1,\ 1)$
$(B_6,\ n_6^1,\ n_6^2)$	0.6	0.4

TABLE XII

TRANSITION PROBABILITY $p(s_l'|s_l)$ WHEN $B_b = B_2$

$p(s_l'\|s_l)$ s_l' s_l	$(B_4,\ n_2^+,\ 1)$
$(B_7,\ 1,\ n_2)$	1

in Tables VII-XII. Transition probabilities in B_3 and B_8 are similar to the ones in Table VIII and Table XI respectively. Likewise, the transition probabilities of B_9, B_{10} and B_{11} are similar to the ones in Table XII. In all of these tables, we use n_i^+ and n_+^{min} to indicate $n_i + 1$ and $n^{min} + 3$ respectively. The probabilities were chosen such that there are no transitions during the minimum green times, the transition probability increases during the rest of the green time, and is 1 when the maximum green time has been reached.

REFERENCES

[1] J. Pucher and R. Buehler, "Making cycling irresistible: Lessons from The Netherlands, denmark and germany," *Transp. Rev.*, vol. 28, no. 4, pp. 495–528, Jul. 2008.

[2] A. Dabiri, A. Hegyi, and B. Goñi-Ros, "Optimal speed advice for cyclists using a roadside sign at signalized intersections with uncertainty in traffic light timing," *Transp. Res. Record: J. Transp. Res. Board*, vol. 2673, no. 7, pp. 239–247, Jul. 2019.

[3] *Cobi Bike*. Accessed: Feb. 15, 2020. [Online]. Available: https://www.bosch-ebike.com/en/products/cobibike/

[4] *Smarthalo*. Accessed: Feb. 20, 2020. [Online]. Available: https://www.smarthalo.bike/

[5] *Gesture Bike*. Accessed: Feb. 20, 2020. [Online]. Available: https://alexandrudancu.com/gesture-bike/

[6] *Digitsole*. Accessed: Feb. 20, 2020. [Online]. Available: https://www.digitsole.com/connected-cycling-insoles-sport-profiler/

[7] J. Andres, T. Kari, J. von Kaenel, and F. Mueller, "Co-riding with my bike to get green Lights," in *Proc. Des. Interact. Syst. Conf.*, Jun. 2019, pp. 1251–1263.

[8] S. Sweeney, R. Ordonez-Hurtado, F. Pilla, G. Russo, D. Timoney, and R. Shorten, "A context-aware E-Bike system to reduce pollution inhalation while cycling," *IEEE Trans. Intell. Transp. Syst.*, vol. 20, no. 2, pp. 704–715, Feb. 2019.

[9] A. Mannion, H. Lhachemi, G. Russo, S. Sweeney, and R. Shorten, "On the design of cyber-physical control system for a smart pedelec (Ebike)," in *Proc. IEEE 58th Conf. Decision Control (CDC)*, Dec. 2019, pp. 2108–2113.

[10] F. Rupi and J. Schweizer, "Evaluating cyclist patterns using GPS data from smartphones," *IET Intell. Transp. Syst.*, vol. 12, no. 4, pp. 279–285, May 2018.

[11] A. Dabiri and A. Hegyi, "Personalised optimal speed advice to cyclists approaching an intersection with uncertain green time," in *Proc. Eur. Control Conf. (ECC)*, Jun. 2018, pp. 1666–1671.

[12] *Flo*. Accessed: Feb. 15, 2020. [Online]. Available: https://bikeflo.com/

[13] B. Asadi and A. Vahidi, "Predictive cruise control: Utilizing upcoming traffic signal information for improving fuel economy and reducing trip time," *IEEE Trans. Control Syst. Technol.*, vol. 19, no. 3, pp. 707–714, May 2011.

[14] K. Yu, J. Yang, and D. Yamaguchi, "Model predictive control for hybrid vehicle ecological driving using traffic signal and road slope information," *Control Theory Technol.*, vol. 13, no. 1, pp. 17–28, Feb. 2015.

[15] B. HomChaudhuri, A. Vahidi, and P. Pisu, "Fast model predictive control-based fuel efficient control strategy for a group of connected vehicles in urban road conditions," *IEEE Trans. Control Syst. Technol.*, vol. 25, no. 2, pp. 760–767, Mar. 2017.

[16] G. Mahler and A. Vahidi, "An optimal velocity-planning scheme for vehicle energy efficiency through probabilistic prediction of traffic-signal timing," *IEEE Trans. Intell. Transp. Syst.*, vol. 15, no. 6, pp. 2516–2523, Dec. 2014.

[17] M. Miyatake, M. Kuriyama, and Y. Takeda, "Theoretical study on eco-driving technique for an electric vehicle considering traffic signals," in *Proc. IEEE 9th Int. Conf. Power Electron. Drive Syst.*, Dec. 2011, pp. 733–738.

[18] G. De Nunzio, C. C. de Wit, P. Moulin, and D. Di Domenico, "Eco-driving in urban traffic networks using traffic signals information," *Int. J. Robust Nonlinear Control*, vol. 26, no. 6, pp. 1307–1324, Apr. 2016.

[19] H. Yang, H. Rakha, and M. Venkat Ala, "Eco-cooperative adaptive cruise control at signalized intersections considering queue effects," *IEEE Trans. Intell. Transp. Syst.*, vol. 18, no. 6, pp. 1575–1585, Jun. 2017.

[20] O. D. Altan, G. Wu, M. J. Barth, K. Boriboonsomsin, and J. A. Stark, "GlidePath: Eco-friendly automated approach and departure at signalized intersections," *IEEE Trans. Intell. Vehicles*, vol. 2, no. 4, pp. 266–277, Dec. 2017.

[21] H. Chen, H. A. Rakha, A. Loulizi, I. El-Shawarby, and M. H. Almannaa, "Development and preliminary field testing of an in-vehicle eco-speed control system in the vicinity of signalized intersections," *IFAC-PapersOnLine*, vol. 49, no. 3, pp. 249–254, 2016.

[22] M. H. Almannaa, H. Chen, H. A. Rakha, A. Loulizi, and I. El-Shawarby, "Reducing vehicle fuel consumption and delay at signalized intersections: Controlled-field evaluation of effectiveness of Infrastructure-to-Vehicle communication," *Transp. Res. Rec., J. Transp. Res. Board*, vol. 2621, no. 1, pp. 10–20, Jan. 2017.

[23] B. HomChaudhuri and P. Pisu, "A control strategy for driver specific driver assistant system to improve fuel economy of connected vehicles in urban roads," in *Proc. Amer. Control Conf. (ACC)*, Jul. 2019, pp. 5557–5562.

[24] S. Yu, R. Fu, Y. Guo, Q. Xin, and Z. Shi, "Consensus and optimal speed advisory model for mixed traffic at an isolated signalized intersection," *Phys. A, Stat. Mech. Appl.*, vol. 531, Oct. 2019, Art. no. 121789.

[25] Y. Feng, D. He, and Y. Guan, "Composite platoon trajectory planning strategy for intersection throughput maximization," *IEEE Trans. Veh. Technol.*, vol. 68, no. 7, pp. 6305–6319, Jul. 2019.

[26] Z. Wang, G. Wu, and M. J. Barth, "Cooperative eco-driving at signalized intersections in a partially connected and automated vehicle environment," *IEEE Trans. Intell. Transp. Syst.*, vol. 21, no. 5, pp. 2029–2038, May 2020.

[27] W. Zhao, D. Ngoduy, S. Shepherd, R. Liu, and M. Papageorgiou, "A platoon based cooperative eco-driving model for mixed automated and human-driven vehicles at a signalised intersection," *Transp. Res. C, Emerg. Technol.*, vol. 95, pp. 802–821, Oct. 2018.

[28] 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.

[29] Z. Wang, G. Wu, P. Hao, and M. J. Barth, "Cluster-wise cooperative eco-approach and departure application for connected and automated vehicles along signalized arterials," *IEEE Trans. Intell. Vehicles*, vol. 3, no. 4, pp. 404–413, Dec. 2018.

[30] Y. Luo, S. Li, S. Zhang, Z. Qin, and K. Li, "Green light optimal speed advisory for hybrid electric vehicles," *Mech. Syst. Signal Process.*, vol. 87, pp. 30–44, Mar. 2017.

[31] Z. Yang, Y. Feng, X. Gong, D. Zhao, and J. Sun, "Eco-trajectory planning with consideration of queue along congested corridor for hybrid electric vehicles," *Transp. Res. Rec., J. Transp. Res. Board*, vol. 2673, no. 9, pp. 277–286, Sep. 2019.

[32] X. Wu, X. He, G. Yu, A. Harmandayan, and Y. Wang, "Energy-optimal speed control for electric vehicles on signalized arterials," *IEEE Trans. Intell. Transp. Syst.*, vol. 16, no. 5, pp. 2786–2796, Oct. 2015.

[33] J. Parkin and J. Rotheram, "Design speeds and acceleration characteristics of bicycle traffic for use in planning, design and appraisal," *Transp. Policy*, vol. 17, no. 5, pp. 335–341, Sep. 2010.

[34] F. Ashtiani, V. S. M. Sreedhara, A. Vahidi, R. Hutchison, and G. Mocko, "Experimental modeling of cyclists fatigue and recovery dynamics enabling optimal pacing in a time trial," in *Proc. Amer. Control Conf. (ACC)*, Jul. 2019, pp. 5083–5088.

[35] M. L. Puterman, *Markov Decision Processes: Discrete Stochastic Dynamic Programming*. Hoboken, NJ, USA: Wiley, 2008.

[36] A. Dabiri, A. Hegyi, and S. Hoogendoorn, "Automatic learning of cyclist's compliance for speed advice at intersections–a reinforcement learning-based approach*," in *Proc. IEEE Intell. Transp. Syst. Conf. (ITSC)*, Oct. 2019, pp. 2375–2380.

[37] Y. Gu, M. Liu, M. Souza, and R. N. Shorten, "On the design of an intelligent speed advisory system for cyclists," in *Proc. 21st Int. Conf. Intell. Transp. Syst. (ITSC)*, Nov. 2018, pp. 3892–3897.

Azita Dabiri received the Ph.D. degree from the Automatic Control Group, Chalmers University of Technology, in 2016. She was a Post-Doctoral Researcher with the Department of Transport and Planning, TU Delft, from 2017 to 2019. In 2019, she received an ERCIM Fellowship and also a Marie Curie Individual Fellowship which allowed her to perform research at the Norwegian University of Technology (NTNU) as a Post-Doctoral Researcher from 2019 to 2020, before joining the Delft Center for Systems and Control, TU Delft, as an Assistant Professor. Her research interests are in the area of integration of model-based and learning-based control and its applications in transportation networks.

Andreas Hegyi (Member, IEEE) received the M.Sc. degree in electrical engineering and the Ph.D. degree from TU Delft, The Netherlands, in 1998 and 2004, respectively. From 2004 to 2007, he was a Post-Doctoral Researcher with TU Delft and with Ghent University. He is currently an Assistant Professor with TU Delft. He is the author or coauthor of more than 100 articles. His research interests are in the areas of traffic flow modeling and control, connected and cooperative vehicles, traffic state estimation, and traffic data analysis. He is a member of IFAC-CTS and has served as the Program Chair of the IEEE-ITSC 2013 conference and as the General Chair of the IXth TRISTAN symposium 2016, and a IPC member of various other conferences. He is an Associate Editor of the IEEE TRANSACTIONS ON INTELLIGENT TRANSPORTATION SYSTEMS and a member of the Editorial Board of Transportation Research Part C.

Serge Hoogendoorn was appointed as an Antonie van Leeuwenhoek Professor Traffic Operations and Management in 2006. He has been the Chair of the Department Transport and Planning since 2018. He is currently (one of the four) a Distinguished Professor of Smart Urban Mobility with the Delft University of Technology. He has a part time appointment with Monash University, and is also an Honorary Professor with the School of Transportation, Southeast University, China. He is also the PI Mobility with the Institute of Advanced Metropolitan Solutions. His current research involves around Smart Urban Mobility, with focal areas, such as theory, modeling, and simulation of traffic and transportation networks, including cars, pedestrian, cyclists and novel public transport services, development of methods for integrated management of these networks, impact of uncertainty of travel behavior and network operations, impact of ICT on network flow operations, robustness, and resilience, and urban data and its applications. In all these topics, his work considers both recurrent and non-recurrent situations.