Available online at www.sciencedirect.com

ScienceDirect

IFAC PapersOnLine 54-2 (2021) 298–305

Vehicle-Based Trajectory Specification in Presence of Traffic Lights with Stochastic Switching Times

P. Typaldos * **V. Volakakis** * **M. Papageorgiou** * **I. Papamichail** *

* *Dynamic Systems & Simulation Laboratory, School of Production Engineering & Management, Technical University of Crete, 73100, Chania, Greece*
(e-mail: {ptypaldos, vvolakakis, markos, ipapa}@ dssl.tuc.gr).

Abstract: Vehicle-based GLOSA (Green Light Optimal Speed Advisory) systems use information about the next switching time of the traffic lights to calculate fuel-efficient position and velocity profiles for connected vehicles, according to their current state (position and speed). A stochastic optimal control problem was recently proposed to address the GLOSA problem in cases where the next switching time is decided in real time and is therefore uncertain in advance. The corresponding numerical solution via SDP (Stochastic Dynamic Programming) calls for substantial computational time (few minutes), which excludes problem solution in the vehicle's computer in real time. This work considers the same stochastic problem of optimal trajectory specification for vehicles approaching a signalized junction with traffic signals operated in real-time (adaptive) mode, due to which the next switching time is stochastic. However, a modified version of Dynamic Programming, known as Discrete Differential Dynamic Programming (DDDP), is used for numerical solution of the stochastic optimal control problem. It is demonstrated, based on a realistic example, that the DDDP algorithm achieves results equivalent to those obtained with the ordinary SDP algorithm, albeit with significantly better performance in terms of computational time. Specifically, the solution is typically obtained in around 1 CPUs, which is real-time feasible and would allow for the DDDP calculations to be executed in the vehicle's on-board computer.

Copyright © 2021 The Authors. This is an open access article under the CC BY-NC-ND license (http://creativecommons.org/licenses/by-nc-nd/4.0)

Keywords: GLOSA, Stochastic Dynamic Programming, Discrete Differential Dynamic Programming (DDDP), Traffic Light Advisory, Speed Advisory

1. INTRODUCTION

In view of cheap energy resources shortage and excessive environmental pollution, it is essential for transportation systems to operate with increased fuel efficiency. In the case of road vehicles, fuel efficiency relates to economic aspects, as fuel economy means fewer expenses for the driver; but also to the protection of the environment in an era of climate crisis. To this end, considerable efforts in the development and deployment of efficient intelligent transportation systems (including real-time traffic signals) lead to reduced congestion and fuel consumption.

Traffic signals guarantee, in the first place, the safe crossing of vehicles at urban junctions in cities around the world. Clearly, enforcing safety via traffic lights implies that some vehicles will have to stop in front of a red light and then accelerate after the traffic light switching to green, something that affects the fuel consumption of concerned vehicles. To reduce the resulting vehicle delays and number of stops, several algorithms have been proposed and deployed over the past decades, aiming at optimizing the traffic signals operation (Hounsell and McDonald (2001); Papageorgiou et al. (2003)). In fact, fuel consumption is increasingly considered as an optimization or evaluation criterion while developing and deploying signal control systems (Jamshidnejad et al. (2017)).

Fixed-time signal plans are derived offline for respective times of day (e.g., morning peak period, off-peak period, etc.) by use of appropriate optimization tools, based on historical constant demands; and are applied without deviations. This implies that switching times of the traffic lights are always known in advance.

In contrast, real-time (or traffic-responsive or adaptive) signal control strategies make use of real-time measurements to calculate in real time suitable signal settings. Depending on the employed signal control strategy, the control update period may range from one second to one signal cycle. Clearly, for real-time signals, the next switching time is not known before the switching decision has been actually made.

Consider a vehicle approaching a red traffic light at a given speed. A common dilemma is whether it should maintain its speed, at the risk of having to stop if the traffic light is still red at arrival; or whether it should decelerate, as long as the traffic light is red, at some uncertain pace. This dilemma of vehicle movement when facing a red traffic signal may be addressed by appropriately designed

2405-8963 Copyright © 2021 The Authors. This is an open access article under the CC BY-NC-ND license.
Peer review under responsibility of International Federation of Automatic Control.
10.1016/j.ifacol.2021.06.035

systems. With recent and emerging advances in vehicle communications, the current state and timing of a traffic signal can be transmitted to equipped vehicles (or apps therein) to enable sensible approaching speed decisions. Based on this information, it is possible to guide the driver (or an automated vehicle) all the way to the traffic light by giving speed advice, which ensures that the vehicle will cross the traffic signal at green and with minimum fuel consumption and emissions. Systems (or apps) optimizing the vehicle approach to traffic lights are often referred to as Green Light Optimal Speed Advisory (GLOSA) systems (Stahlmann et al. (2016)).

In the case of fixed signals and hence prior knowledge of the next switching time, a corresponding message is broadcasted by the signal controller. Under these conditions, the problem of how to optimize the approach to traffic signals has been addressed in different ways. Rule-based algorithms have been proposed in various works to produce advisory speeds for vehicles approaching traffic signals, so as to reduce fuel consumption and emissions (Katsaros et al. (2011); Sanchez et al. (2006); Ma et al. (2018)). Optimal control approaches, considering explicitly the vehicle kinematics, are, by their nature, more pertinent in producing fuel-optimal speed profiles (Lawitzky et al. (2013); Typaldos et al. (2020a)).

The situation becomes more complicated when real-time signals with very short (e.g., second-by-second) control update periods are present, in which case exact prior knowledge of the next switching time is not available. In this case, the best available knowledge can be presented as an estimate (Koukoumidis et al. (2011)) or as a probabilistic distribution for the next switching time within a short-term future time-window; such a distribution may be obtained by use of statistics from previous signal operation (Mahler and Vahidi (2012); Lawitzky et al. (2013)).

In Typaldos et al. (2020a), the problem of producing fuel-optimal vehicle trajectories for a vehicle approaching a traffic signal for both cases of known and stochastic switching times was considered. For the first case, the problem was formulated as an optimal control problem and was solved analytically via PMP (Pontryagin's Maximum Principle). Subsequently, the case of stochastic switching time with known probability distribution was also addressed, and the problem was cast in the format of a stochastic optimal control problem, which was solved numerically using SDP (Stochastic Dynamic Programming). The proposed SDP algorithm may take several minutes to execute, which implies that the solution is not real-time feasible and can therefore not be obtained on-board the vehicle, but must be executed offline, at the infrastructure side, and be communicated to approaching vehicles according to their current state.

To substantially reduce the computation time and memory requirements for the solution of the above-mentioned stochastic GLOSA problem and enable its solution on-board the vehicle, it is employed, in the current work, a Discrete Differential Dynamic Programming (DDDP) algorithm. DDDP was proposed by Heidari et al. (1971) for deterministic problems in the context of water resources system optimisation and has been widely used in that domain to reduce the computational requirements compared with the standard DP (Dynamic Programming) algorithm of Bellman (2015). DDDP is an iterative algorithm, whereby each iteration receives a feasible (but non-optimal) solution trajectory and transforms it to an enhanced one, to be used in the next iteration. The procedure starts, at the first iteration, with a feasible starting trajectory provided by the user. Each iteration employs the conventional DP algorithm to solve the problem within a strongly reduced state domain (compared to the original problem's state domain) around the received state trajectory of the previous iteration. The procedure stops, when, at some iteration, the received trajectory is found to be the optimal one, hence it cannot be further enhanced. It is also possible to reduce the discretisation intervals, while advancing with the iterations, in order to improve the solution accuracy. The computational time required for each iteration is far lower compared to that of the one-shot full problem solution due to the strongly reduced space domain considered. Thus, even though we have multiple successive iterations, the computational time for all of them may still be significantly lower than for the one-shot full problem solution.

It should be noted that, in contrast to deterministic optimal control problems, stochastic optimal control problems do not feature a solution trajectory, even for given initial states, due to the uncertainty of system evolution created by the stochastic variables. However, in the specific stochastic problem (stochastic GLOSA) considered here, such a trajectory is indeed present in the problem solution, and this allows for application of the DDDP algorithm despite the stochastic nature of the problem.

The remainder of the paper is organized as follows: in Section 2, the optimal control problems with known signal switching time and uncertain signal switching time, as proposed by Typaldos et al. (2020a), are briefly presented for completeness, followed by the presentation of the DDDP algorithm. Demonstration results of the DDDP algorithms performance and comparison with the one-shot stochastic GLOSA results, are presented in Section 3. Finally, Section 4 concludes this work, summarising its contributions.

2. OPTIMAL CONTROL PROBLEMS AND SOLUTIONS FOR GLOSA WITH KNOWN OR UNKNOWN SIGNAL SWITCHING TIME

This section describes briefly the GLOSA approaches for the cases of known or unknown signal switching time, as proposed by Typaldos et al. (2020a); followed by the description of the DDDP approach proposed in this paper.

2.1 Known Signal Switching Time

Consider a vehicle traveling from an initial state $\boldsymbol{x}_0 = [x_0, v_0]^T$, with x_0 being a given initial position and v_0 a given initial speed of the vehicle; with the purpose to reach a fixed final state $\boldsymbol{x}_e = [x_e, v_e]^T$ within a free (but weighted) time horizon t_e, with x_e and v_e being the vehicle's final position and speed, respectively. Between the initial and final positions, there is a traffic signal, and hence the additional restriction that the vehicle cannot pass through the traffic signal's position x_1 before the known time t_1, which is the time that the traffic light turns green from red. The implicit assumption here is that a red light is active when the vehicle appears on the link (at time 0), but a generalisation, which includes the case where the vehicle appears when the traffic light is green, is given in Typaldos et al. (2020a). The objective of the vehicle is to appropriately adjust its acceleration (control variable), so as to minimize fuel consumption, while satisfying the initial and final conditions $\boldsymbol{x}_0$ and $\boldsymbol{x}_e$, as well as the intermediate (traffic signal) constraint.

The minimization problem outlined above is formulated as an optimal control problem, which accounts for the vehicle kinematics via the following state equations:

$$\dot{x} = v \tag{1}$$

$$\dot{v} = a \tag{2}$$

where a is the vehicle acceleration which assumes the role of the control variable. The objective criterion to be minimized reads

$$J = w \cdot t_e + \frac{1}{2}\int_0^{t_e} a^2 dt \tag{3}$$

In addition, the green-light constraint, $t_S > t_1$, must be fulfilled, where t_S is the time at which the vehicle crosses from the signal position x_1, that is, $x(t_S) = x_1$. Note that the utilized acceleration cost term a^2 in the cost criterion was demonstrated in an earlier study to be an excellent proxy for deriving fuel-minimizing vehicle trajectories Typaldos et al. (2020b). Note also that the final time t_e is free, but penalized with the parameter w. For higher values of w, the resulting t_e will be lower and vice versa. This, consequently, affects the acceleration cost, which, depending on higher or lower w value, will also have increased or decreased values (for more details see Typaldos et al. (2020a)). If necessary, upper and lower bounds may be applied to speed v and acceleration a.

The solution of this problem was addressed in Typaldos et al. (2020a) and can be obtained analytically using symbolic differentiation tools. Thus, the numerical solution of the deterministic GLOSA problem, for a specific problem instance, takes only fractions of a second of computation time and can be executed in real time on-board for each approaching vehicle.

For a given junction, the final state is the same for any initial vehicle state $\boldsymbol{x}_0$ and any switching time t_1. Therefore, the optimal value of criterion (3) of the deterministic GLOSA problem depends on these variables and is denoted $J^*_{DG}(\boldsymbol{x}_0, t_1)$ for later use.

2.2 Uncertain Signal Switching Time Problem

The traffic light switching time may be subject to short-term decisions in dependence of the prevailing traffic conditions in cases of real-time signals. In such cases, we typically have minimum and maximum admissible switching times; hence, based on statistics from past signal switching activity, we may derive a probability distribution of switching times within the admissible time-window of possible signal switching times. Thus, the problem can be cast in the format of a stochastic optimal control problem, which may be solved numerically using SDP techniques. To this end, the analytical solution of the deterministic GLOSA optimal control problem is used within the stochastic approach, as will be explained in this section (for more details see Typaldos et al. (2020a)).

For the SDP algorithm (Bertsekas (1995)), the discrete-time version of the vehicle kinematics, with time step T, is considered, as follows:

$$x(k+1) = x(k) + v(k)T + \frac{1}{2}a(k)T^2 \tag{4}$$

$$v(k+1) = v(k) + a(k)T \tag{5}$$

where $x(k), v(k)$ correspond to the vehicle position and speed at discrete times $k = 0, 1, \ldots$ (where $kT = t$), while the control variable $a(k)$ reflects the acceleration that remains constant over each time period k. The state and control variables are bounded within the following feasible regions

$$\boldsymbol{x}(k) \in \boldsymbol{X} = [\boldsymbol{x}_{\min}, \boldsymbol{x}_{\max}] \tag{6}$$

$$a(k) \in U = [a_{\min}, a_{\max}] \tag{7}$$

with $\boldsymbol{x}_{\min}, \boldsymbol{x}_{\max}$ and $a_{\min}, a_{\max}$ being the lower and upper bounds of the states and acceleration, respectively. The traffic light's discrete switching time k_1 is not known, but it is assumed that a known range $k_{\min} \leq k_1 \leq k_{\max}$ of possible switching times exists, with $k_{\min}$ and $k_{\max}$ being the minimum and maximum possible switching times.

For proper problem formulation, a virtual state $\tilde{x}(k)$ is introduced, that reflects formally the stochasticity of traffic light switching

$$\begin{aligned} \tilde{x}(k+1) &= \tilde{x}(k) \cdot z(k) \\ \tilde{x}(0) &= 1 \end{aligned} \tag{8}$$

where $z(k)$ is a binary stochastic variable defined as

$$z(k) = \begin{cases} 0 & \text{if traffic light switches at time } k+1 \\ 1 & \text{else} \end{cases} \tag{9}$$

with (8) and (9), the virtual state $\tilde{x}(k)$ is either equal to 1, if the traffic light has not yet switched until time $k-1$; or equal to 0 if switching occurred at time k or earlier. The virtual state $\tilde{x}$ is assumed measurable, which means that the system knows at each time kT if switching has taken place or not within the last time period $[(k-1)T, kT]$.

The stochastic variable $z(k)$ is independent of its previous values and takes values according to a time-dependent probability distribution $p(z|k)$. Based on the statistics of previous signal switching activity, availability of an a-priori discrete probability distribution $P(k), k_{\min} \leq k_1 \leq k_{\max}$ is assumed, for signal switching within the time-window, where $\sum_{k=k_{\min}}^{k_{\max}} P(k) = 1$. Since no switching takes place for $k \leq k_{\min} - 1$, we have

$$p(0|k) = 0 \quad \text{for} \quad k < k_{\min} - 1. \tag{10}$$

For $k \geq k_{\min}$, the probability distribution $p(z|k)$, is obtained by use of "crop-and-scale", meaning that the a-priori probabilities of previous time steps, where switching did not take place, are distributed analogously to increase the probabilities of the remaining discrete times within the time-window (Lawitzky et al. (2013)). As shown by

Typaldos et al. (2020a), this update may be done by use of the following crop-and-scale formula that applies for $k_{\min} \leq k \leq k_{\max} - 1$ and for any a-priori distribution $P(k)$

$$p(0|k) = \frac{P(k+1)}{\sum_{\kappa=k+1}^{k_{\max}} P(\kappa)} \tag{11}$$

where the term in brackets reflects the crop-and-scale update.

The cost criterion of the stochastic problem is the same as in the deterministic case (3). However, in the stochastic case, the exact value of the criterion depends on the stochastic variable's realization, and therefore we consider minimisation of the expected value

$$J = E\Big\{w \cdot t_e + \frac{1}{2}\int_0^{t_e} a^2 dt\Big\} \tag{12}$$

where the expectation refers to the stochastic variable $z(k), k = 0, \ldots, k_{\max} - 1$. Note that, when the switching time becomes known at time k_1, while the vehicle is at state $x(k_1)$, the problem instantly becomes a deterministic GLOSA problem, and the corresponding optimal cost-to-go is $J^*_{DG}[\boldsymbol{x}(k_1), k1]$, which will be denoted as the "escape cost".

To obtain a formally proper cost criterion, the stochastic variable $z(k)$ and the virtual variable $\tilde{x}(k)$ introduced earlier are used, and, as shown by Typaldos et al. (2020a), this yields the objective function in the required form, as follows

$$J = E\Bigg\{\tilde{x}(k) \sum_{k=0}^{k_{\max}-1} \Bigg[\frac{1}{2}a(k)^2 + [1 - z(k)]J^*_{DG}[\boldsymbol{x}(k), a(k), k+1]\Bigg]\Bigg\} \tag{13}$$

Equations (4)-(13) constitute an ordinary stochastic optimal control problem (Bertsekas (1995)). Denoting the corresponding optimal cost-to-go function by $V[\boldsymbol{x}(k), \tilde{x}(k), k]$, the recursive Bellman equation for $0 \leq k \leq k_{\max} - 1$ reads

$$\begin{aligned} V[\boldsymbol{x}(k), \tilde{x}(k), k] &= \\ &= \min_{a(k)\in U}\Bigg\{E\Big\{\frac{1}{2}a(k)^2 + [1 - z(k)]J^*_{DG}[\boldsymbol{x}(k), a(k), k+1] \\ &\quad + V[\boldsymbol{x}(k+1), \tilde{x}(k)z(k), k+1]\Big\}\Bigg\} \\ &= \min_{a(k)\in U}\Bigg\{\frac{1}{2}a(k)^2 + p(0|k)\cdot J^*_{DG}[\boldsymbol{x}(k), a(k), k+1] \\ &\quad + [1 - p(0|k)]\cdot V[\boldsymbol{x}(k+1), 1, k+1]\Bigg\} \end{aligned} \tag{14}$$

with boundary condition $V[\boldsymbol{x}(k_{\max}), 1, k_{\max}] = 0$.

2.3 Discrete SDP Numerical Solution Algorithm

For the numerical solution of the stochastic problem, using the SDP algorithm, the state and control variables must be discretised. As the discretization level has a significant impact on computational time and memory requirements, but also on the accuracy of the computed solution, an appropriate trade-off should be specified between reasonable computation requirements versus achievable solution quality.

For the discretization, the discrete time interval, T is set equal to 1 s, which is a reasonable choice for the problem at hand. Then, a general discretization interval Δ for the problem variables is assumed, and the discretization interval of acceleration is set $\Delta a = \Delta$. From (5), the discretization interval of speed assumes the same value ($\Delta v = \Delta a = \Delta$). Likewise, in view of (4), the discretization interval for the position is

$$\Delta x = \frac{1}{2}\Delta \cdot T^2. \tag{15}$$

Based on these settings, it was shown in Typaldos et al. (2020a) that, if $x(k), v(k), a(k)$ are discrete points, then $x(k+1)$ and $v(k+1)$ resulting from (4) and (5) are also discrete points.

It is now straightforward to apply the discrete SDP algorithm to obtain an optimal closed-loop control law $a(k)^* = R[\boldsymbol{x}(k), k]$, which for any given vehicle state $\boldsymbol{x}(k) \in X$ and time k, delivers the optimal acceleration $a(k)^*$. The SDP algorithmic steps are summarized below. Note that the algorithm needs to consider only the case $\tilde{x}(k) = 1$, therefore any arguments pertaining to $\tilde{x}(k)$ are suppressed for convenience.

The SDP algorithm is described as follows:

$V[\boldsymbol{x}(k_{\max}), k_{\max})] \leftarrow 0 \quad \forall \boldsymbol{x}(k_{\max}) \in \boldsymbol{X}$
for each $k = k_{\max} - 1, \ldots, 0$ **do**
  for each discrete state $\boldsymbol{x}(k) \in \boldsymbol{X}$ **do**
    for each discrete control $a(k) \in U$ **do**
      Calculate $x(k+1), v(k+1)$
      if $\boldsymbol{x}(k+1) \notin \boldsymbol{X}$ **then**
        $J[\boldsymbol{x}(k), a(k), k] \leftarrow \infty$
        continue
      end if
      $J[\boldsymbol{x}(k), a(k), k] \leftarrow \frac{1}{2}a(k)^2 +$
        $p(0|k) \cdot J^*_{DG}[\boldsymbol{x}(k), a(k), k] +$
        $[1 - p(0|k)] \cdot V[\boldsymbol{x}(k+1), k+1]$
    end for
    $V[\boldsymbol{x}(k), k] \leftarrow \min J[\boldsymbol{x}(k), a(k), k] \quad \forall a(k) \in U$
    $R[\boldsymbol{x}(k), k] = a(k)^* \leftarrow \arg\min_{a(k)\in U}\{J[\boldsymbol{x}(k), a(k), k]\}$,
    with $a(k)^*$ the optimal control of point $[\boldsymbol{x}(k), k]$
  end for
end for

As mentioned, this algorithm delivers an optimal control law $a(k)^* = R[\boldsymbol{x}(k), k]$ for the full state domain $\boldsymbol{X}$. Note that general stochastic optimal control problems do not possess a solution trajectory for specific initial states $\boldsymbol{x}_0$, because the state evolution is uncertain in presence of the stochastic variables. However, in the specific GLOSA problem addressed here, the evolution of the vehicle state (4), (5) is not affected by the stochastic variable $z(k)$, which concerns only the signal switching time. Thus, for a given initial state, i.e. vehicle position and speed at time 0, the optimal control law may be used to produce an optimal vehicle trajectory that the vehicle should pursue; until the signal switching actually occurs, in which case the vehicle should instantly behave according to the deterministic GLOSA solution.

2.4 Discrete Differential Dynamic Programming

The major disadvantage of the discrete (S)DP algorithm is the high computation time required for the numerical solution of the optimal control problem. To address this weakness, several modified DP algorithms have been proposed, which lead to reduction of the computational effort; and one of them is the DDDP algorithm, proposed in Heidari et al. (1971) for deterministic optimal control problems. The method can nevertheless be applied here, because an optimal vehicle trajectory may be derived for the stochastic GLOSA problem, despite its stochastic character.

As already mentioned, DDDP is an iterative algorithm, calling for a feasible starting state trajectory to be specified externally. Each iteration l receives a feasible (but non-optimal) state trajectory $\boldsymbol{x}^{(l-1)}(k)$, and transforms it to an enhanced one $\boldsymbol{x}^{(l)}(k)$, to be used in the next iteration. To this end, each iteration solves a discrete SDP problem by use of the standard SDP algorithm presented above. What changes at each iteration l is the considered state domain $\boldsymbol{X}_C^{(l)} = \{\boldsymbol{x}(k) \mid |\boldsymbol{x}(k) - \boldsymbol{x}^{(l-1)}(k)| \leq \boldsymbol{\Delta}_C^{(l)} \wedge \boldsymbol{x}(k) \in \boldsymbol{X}\}$, which is a strongly reduced subdomain of the original state domain $\boldsymbol{X}$ in (6). In other words, the discretized SDP problem is solved in each iteration l within a corridor with width $\boldsymbol{\Delta}_C^{(l)}$ around the received state trajectory $\boldsymbol{x}^{(l-1)}(k)$ of the previous iteration, to produce a solution trajectory $\boldsymbol{x}^{(l)}(k)$ for use in the next iteration. The corridor width $\boldsymbol{\Delta}_C^{(l)}$, as well as the discretisation intervals $\Delta\boldsymbol{a}^{(l)}, \Delta\boldsymbol{x}^{(l)}$, can vary in each iteration, typically at a decreasing rate. The procedure stops when termination criterion is satisfied.

In the proposed GLOSA application, the starting trajectory for the first iteration of DDDP is the optimal solution of the deterministic GLOSA problem, assuming the "pessimistic" case where the traffic light will switch from red to green at the latest possible time, that is, at $k_1 = k_{\max}$, so as to cover the whole time range and be not too far from the stochastic optimal trajectory; see Typaldos et al. (2020a). The discretisation intervals $\Delta a^{(1)}, \Delta\boldsymbol{x}^{(1)}$ for the first iteration are also given. The intervals are reduced by half, each time there is no solution improvement at two subsequent iterations. The corridor width is taken as $\boldsymbol{\Delta}_C^{(l)} = (\boldsymbol{C} \cdot \Delta a^{(l)})$, where $\boldsymbol{C} = [\boldsymbol{C}_x, \boldsymbol{C}_v]$ are constant given values specifying the state corridor width. Thus, the corridor's initial width is defined through the chosen values for $\boldsymbol{C}$ and $\Delta a^{(l)}$. In the following iterations, whenever the discrete interval is reduced, there is an analogous reduction of the corridor width. Consequently, we have a constant number of feasible discrete points in all iterations, which facilitates the algorithm's fine-tuning so as to improve its computational efficiency. Note that, $\boldsymbol{C}_x$ and $\boldsymbol{C}_v$ values may differ, as the magnitude of the two respective state variables differs.

The admissible control region U (see (7)) is kept the same at each iteration, although many related state transitions cannot be considered in view of the reduced state variables domain. The algorithm terminates whenever there is no improvement of the produced solution trajectory even after a reduction of the discretisation intervals; or when the $\Delta a^{(l)}$ value becomes less than 0.125, which was found in Typaldos et al. (2020a) to lead to sufficiently accurate results.

It should be noted that there is no general guarantee that the DDDP iterations will actually converge to the full-domain SDP solution. In particular, if the state sub-domains $\boldsymbol{X}_C^{(l)}$ considered in the iterations are too small, the obtained DDDP solution may actually differ from the SDP solution. On the other hand, if the state sub-domains are selected large, the required number of iterations may decrease, but the computation time required to find the solution at each iteration increases accordingly. In conclusion, some fine-tuning regarding the size of sub-domains is necessary to ensure convergence to the SDP solution with minimum overall (all iterations) computation time.

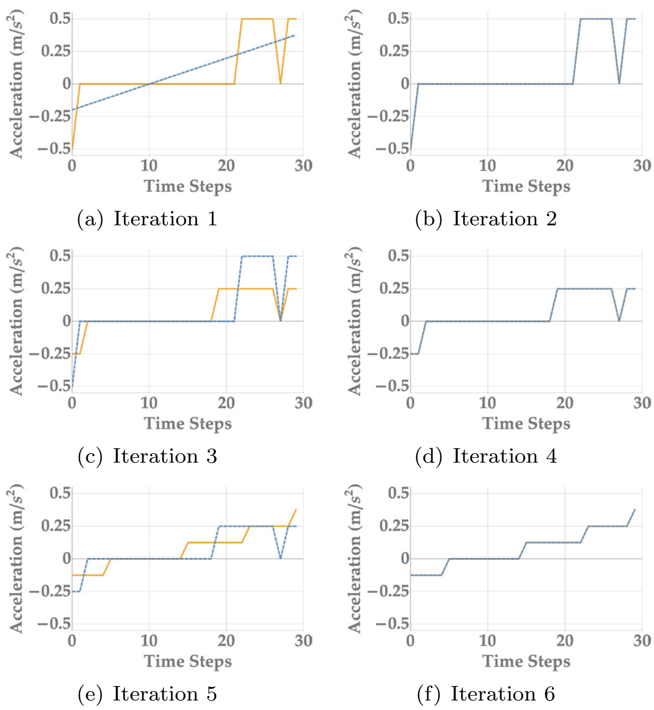
3. RESULTS

In this section, some results using the proposed DDDP approach are presented. Several scenarios have been tested, with different initial and final conditions, different switching windows and different probability distributions. In these scenarios, different situations occur, such as cases where the vehicle needs to accelerate or decelerate before or after crossing the traffic signal; or cases where the vehicle is even forced to fully stop and wait until the traffic light's switch from red to green.

One scenario will be presented here (more results and scenarios are being considered in ongoing work) with the following initial and final conditions: $x_0 = 0\,\mathrm{m}$, $v_0 = 5\,\mathrm{m/s}$, $x_e = 220\,\mathrm{m}$, $v_e = 11\,\mathrm{m/s}$ and $w = 0.1$. The traffic light position is: $x_1 = 150\,\mathrm{m}$. The states and control bounds are set to $[x_{\min}, x_{\max}] = [0, 150]\,\mathrm{m}$, $[v_{\min}, v_{\max}] = [0, 16]\,\mathrm{m/s}$, $[a_{\min}, a_{\max}] = [-3, 3]\,\mathrm{m/s^2}$, respectively. The time step T is 1 s and the switching time range for the traffic light is $[k_{\min}, k_{\max}] = [10, 30]$ with uniform a-priori probability distribution. For the initial discretization, $\Delta a = \Delta v = 0.5$ is used, which leads to $\Delta x = 0.25\,\mathrm{m}$ and the initial corridor width is set $\boldsymbol{C} = [20, 4]$, i.e. we have $\boldsymbol{C}_x = 5{\cdot}\boldsymbol{C}_v$. The choice of the initial discretization and corridor width value will be justified later in this section.

Fig. 1 and 2 display the optimal state (speed) and control (acceleration) trajectories over each iteration of the DDDP algorithm. Note that, position trajectories are not included, as the difference of those trajectories, at each iteration, is small and barely visible. Specifically, in each iteration we consider a corridor $\boldsymbol{\Delta}_C^{(l)} = [-\boldsymbol{C} \cdot \Delta a^{(l)}, \boldsymbol{C} \cdot \Delta a^{(l)}]$ around the respective received state trajectories, which cannot of course extend out of the full state bounds.

In both Fig. 1 and 2, the dashed blue lines represent, for each iteration, the received trajectory, the solid orange lines represent the optimal trajectories derived, and the red dashed lines reflect the corresponding corridor bounds. It can be observed from Figs. 1-2 that, starting with the initial chosen discretization, the first DDDP iteration improves the initial trajectory, leading to a better solution, which is optimal within the considered sub-domain. In the second iteration, no further improvement can be achieved, which means that, with the current discretization values,



(a) Iteration 1 (b) Iteration 2
(c) Iteration 3 (d) Iteration 4
(e) Iteration 5 (f) Iteration 6

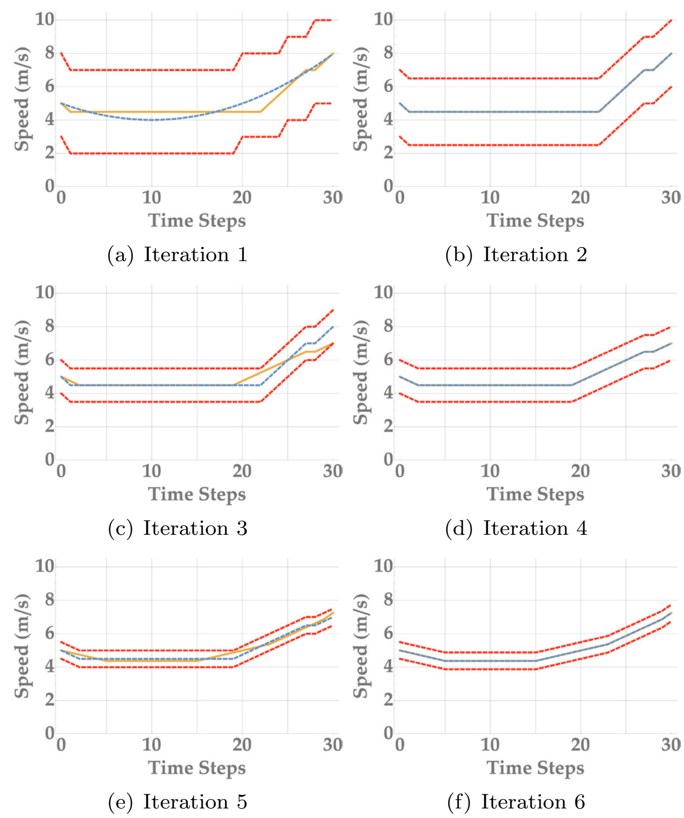
Fig. 1. Received (blue dashed line) and optimal (orange line) acceleration trajectories of DDDP algorithm in each iteration.

the best achievable solution has been reached. By reducing the discretization at the 3rd iteration, a reduction in the corridor width is observed, but the number of discrete points remains the same. This reduction enables further improvement of the solution, and the procedure continuous until the termination criterion is fulfilled. Table 1 contains the values of the objective function for each iteration of the DDDP algorithm, assuming $\Delta a = 0.5$ and $C = [20, 4]$, where the choice of these values will be explained in the following.

Table 1. Optimal cost evolution and descretization change in each iteration of DDDP algorithm.

Iter.	$\Delta a = \Delta v$	Δx	Cost
1	0.5	0.25	1.357
2	0.5	0.25	1.357
3	0.25	0.125	1.223
4	0.25	0.125	1.223
5	0.125	0.0625	1.175
6	0.125	0.0625	1.175

In table 2, the results of DDDP for different initial corridor $\boldsymbol{C}$ and discretization Δa values are presented. It can be seen that, for all presented discretization values, as the corridor's width is increased, a reduction of the number of DDDP iterations is observed. This behaviour is expected, as the bigger corridors, i.e. bigger admissible regions, lead to potentially better solutions at each iteration, and hence to fewer iterations to reach the original SDP problem's optimal solution. On the other hand, despite the decrease on the number of iterations, the overall computation time is increased due to the higher computation time required at each iteration, which in turn, is due to more feasible discrete state points included. Moreover, for very small values of $\boldsymbol{C}$, it is noticed that DDDP could not converge to



(a) Iteration 1 (b) Iteration 2
(c) Iteration 3 (d) Iteration 4
(e) Iteration 5 (f) Iteration 6

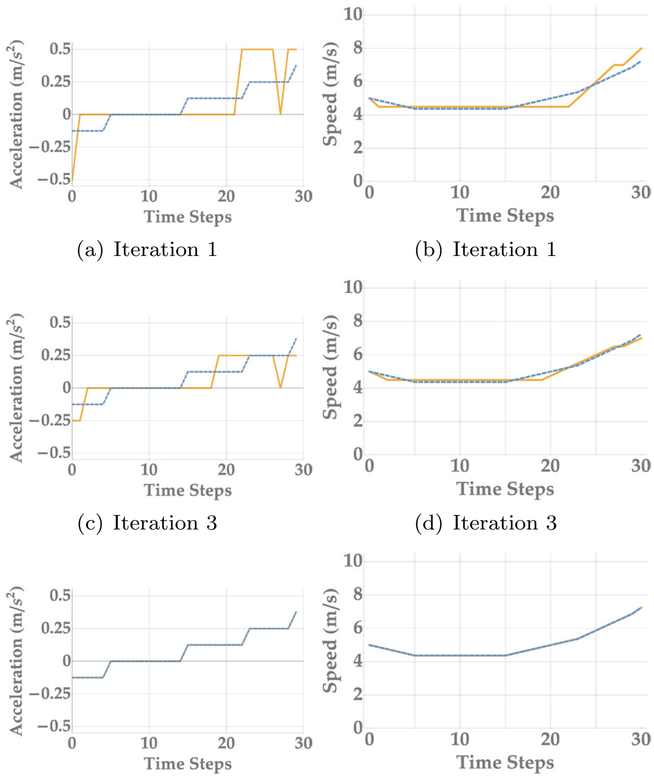
Fig. 2. Received (blue dashed line) and optimal (orange line) speed trajectories of DDDP algorithm in each iteration. The corridor $\boldsymbol{\Delta}_C^{(l)}$ is marked with red dashed lines.

the best possible solution when the termination criterion is fulfilled, due to the extremely limited state space. Based on these observations, the selection of the values $\boldsymbol{C} = [20, 4]$ and $\Delta a = 0.5$ seems to be a reasonable choice, and this choice was found to lead to similar results also in several other scenarios, not presented here.

Table 2. Performance of DDDP algorithm in terms of CPU-time and optimal cost for different initial values of $\boldsymbol{C}$ and Δa.

	$\Delta a = 1.0$			$\Delta a = 0.5$		
C_v	Iter.	CPU time (s)	Cost	Iter.	CPU time (s)	Cost
2	19	0.648	1.199	20	0.612	1.193
3	19	1.296	1.177	8	0.552	1.175
4	15	**1.655**	1.175	6	**0.692**	1.175
5	12	1.839	1.175	6	1.052	1.175
6	10	2.298	1.175	6	1.451	1.175
	$\Delta a = 0.25$			$\Delta a = 0.125$		
C_v	Iter.	CPU time (s)	Cost	Iter.	CPU time (s)	Cost
2	10	0.326	1.193	5	0.280	1.193
3	6	0.469	1.179	5	0.443	1.179
4	7	**0.867**	1.175	5	**0.762**	1.175
5	6	1.111	1.175	4	0.885	1.175
6	5	1.291	1.175	3	0.967	1.175

The accuracy of the DDDP algorithm compared to the full one-shot SDP solution is assessed in Fig. 3. In this figure, the optimal state (speed) and control (acceleration) trajectories (orange lines), in the first, middle and last DDDP iterations are contrasted to the corresponding optimal trajectories derived from the full-range SDP (blue dashed lines) with a discretisation of $\Delta a = 0.125$. It



(a) Iteration 1 (b) Iteration 1

(c) Iteration 3 (d) Iteration 3

(e) Iteration 6 (f) Iteration 6

Fig. 3. Optimal acceleration and speed trajectories (orange line) of three DDDP iterations, compared with corresponding optimal trajectories of the one-shot SDP (blue dashed line).

can be seen that DDDP, starting in the first iteration with the initial trajectory derived from the "pessimistic" deterministic GLOSA problem, manages to converge at the exact same optimal solution as the full-range SDP. The obtained optimal cost of both approaches is 1.17517, while the computation time difference is remarkable, as the one-shot SDP needs 613.86 s to obtain the solution, while DDDP needs only 0.69 s. More importantly, this big reduction in computation time enables the DDDP algorithm to be executable in real time, even in an MPC (Model Predictive Control) mode, on the vehicle side, similarly to the deterministic GLOSA.

4. CONCLUSIONS

The current work is an extension of a previous work (Typaldos et al. (2020a)), where a stochastic GLOSA methodology was developed, by optimizing, using SDP techniques, the vehicle kinematic trajectories subject to the intermediate stochastic traffic signal switching constraint and with a fixed final state. In the present extension, a Discrete Differential Dynamic Programming algorithm (DDDP) algorithm was developed, which solves the original SDP problem iteratively, each time considering a reduced state space. Demonstration results demonstrate that the DDDP algorithm strongly outperforms, by a factor of 1:1000 the full-range SDP in terms of computation time. This enables the DDDP algorithm to be executable in real time onboard approaching vehicles, even in a model predictive control (MPC) mode.

Current and future work is focused on:

- Generalization of the current GLOSA problem by considering uncertain switching times for both green and red phases.
- Solving the SDP problem with a different modified iterative DP algorithm, in an attempt to further reduce the computational time.

ACKNOWLEDGEMENTS

The research leading to these results has received funding from the European Research Council under the European Unions Seventh Framework Programme (FP/2007-2013) / ERC Grant Agreement n. [321132], project TRAMAN21 as well as from related matching funds by the Greek Ministry of Education, Research and Religious Affairs (General Secretariat for Research and Technology).

REFERENCES

Bellman, R.E. (2015). *Adaptive control processes: a guided tour*, volume 2045. Princeton university press.

Bertsekas, D.P. (1995). *Dynamic programming and optimal control*, volume 1. Athena scientific Belmont, MA.

Heidari, M., Chow, V.T., Kokotović, P.V., and Meredith, D.D. (1971). Discrete differential dynamic programing approach to water resources systems optimization. *Water Resources Research*, 7(2), 273–282.

Hounsell, N. and McDonald, M. (2001). Urban network traffic control. *Proceedings of the Institution of Mechanical Engineers, Part I: Journal of Systems and Control Engineering*, 215(4), 325–334.

Jamshidnejad, A., Papamichail, I., Papageorgiou, M., and De Schutter, B. (2017). Sustainable model-predictive control in urban traffic networks: Efficient solution based on general smoothening methods. *IEEE Transactions on Control Systems Technology*, 26(3), 813–827.

Katsaros, K., Kernchen, R., Dianati, M., and Rieck, D. (2011). Performance study of a green light optimized speed advisory (glosa) application using an integrated cooperative its simulation platform. In *2011 7th International Wireless Communications and Mobile Computing Conference*, 918–923. IEEE.

Koukoumidis, E., Peh, L.S., and Martonosi, M.R. (2011). Signalguru: leveraging mobile phones for collaborative traffic signal schedule advisory. In *Proceedings of the 9th international conference on Mobile systems, applications, and services*, 127–140.

Lawitzky, A., Wollherr, D., and Buss, M. (2013). Energy optimal control to approach traffic lights. In *2013 IEEE/RSJ International Conference on Intelligent Robots and Systems*, 4382–4387. IEEE.

Ma, J., Zhou, F., Huang, Z., Melson, C.L., James, R., and Zhang, X. (2018). Hardware-in-the-loop testing of connected and automated vehicle applications: a use case for queue-aware signalized intersection approach and departure. *Transportation Research Record*, 2672(22), 36–46.

Mahler, G. and Vahidi, A. (2012). Reducing idling at red lights based on probabilistic prediction of traffic signal timings. In *2012 American Control Conference (ACC)*, 6557–6562. IEEE.

Papageorgiou, M., Diakaki, C., Dinopoulou, V., Kotsialos, A., and Wang, Y. (2003). Review of road traffic control strategies. *Proceedings of the IEEE*, 91(12), 2043–2067.

Sanchez, M., Cano, J.C., and Kim, D. (2006). Predicting traffic lights to improve urban traffic fuel consumption. In *2006 6th International Conference on ITS Telecommunications*, 331–336. IEEE.

Stahlmann, R., Möller, M., Brauer, A., German, R., and Eckhoff, D. (2016). Technical evaluation of glosa systems and results from the field. In *2016 IEEE Vehicular Networking Conference (VNC)*, 1–8. IEEE.

Typaldos, P., Kalogianni, I., Mountakis, K.S., Papamichail, I., and Papageorgiou, M. (2020a). Vehicle trajectory specification in presence of traffic lights with known or uncertain switching times. *Transportation research record*, 2674(8), 53–66.

Typaldos, P., Papamichail, I., and Papageorgiou, M. (2020b). Minimization of fuel consumption for vehicle trajectories. *IEEE Transactions on Intelligent Transportation Systems*, 21(4), 1716–1727.