The Netherlands
Research School for
Transport, Infrastructure
and Logistics
TRAIL

11th TRAIL Congress
November 2010

FREEWAY TRAFFIC CONTROL USING Q-LEARNING

Mohsen Davarynejad, MSc [1], **Andreas Hegyi, PhD** [2], **Jos Vrancken, PhD** [1], **Yubin Wang, MSc** [1]

[1]Faculty of Technology, Policy and Management, Delft University of Technology, the Netherlands

[2]Faculty of Civil Engineering and Geo Sciences, Department of Transport and Planning, Delft University of Technology, the Netherlands

ABSTRACT

In this paper, the standard Q-learning algorithm is applied to control the density of an arbitrary freeway via ramp metering in a macroscopic level. The reinforcement learning algorithms have proven to be effective tools for letting an agent learn from its experiences generated by its interaction with an environment. The performance of the algorithm as well as its robustness against communication failure is studied. The results of the simulations demonstrated the effectiveness the technique.

KEYWORDS

Freeway traffic control, Ramp metering, Reinforcement learning, Q-learning

INTRODUCTION

In this paper we consider traffic networks that are controlled by ramp metering. When traffic is dense, by limiting the inflow into, ramp metering can prevent a traffic breakdown such that the density remains below the critical value thus avoiding congestion Hegyi et al. (2005). Model Predictive Control Maciejowski (2002) is a technique that is frequently used for the purposes of ramp metering control Bellemans et al. (2003). The fact that MPC is model-based results in risk of predictive controller to misbehave when considering model mismatch. Moreover non-linear optimizations that are part of MPC approaches may fail to find a good solution Ernst et al. (2009). To elude this problem and to avoid the computational complexity of MPC, in this paper, reinforcement learning algorithms are used for ramp metering control problem since techniques of this kind do not require a model of the environment, among other benefits.

In reinforcement learning (RL), while interacting with the environment and through trial and error, each agent learns an efficient policy to reach a goal as fast as possible. RL is an unsupervised active learning process, aiming at developing algorithms for solving sequential decision

problems (SDPs), by which agents learn to achieve goals from their interactions with the environment. In RL, the learner perceives environment state, takes an action and receives a scalar signal providing evaluative information on the quality of the action. The signal does not provide any instructive information on the best behavior in that state. At each time step, the scalar signal (reinforcement signal) can be positive (known as reward), negative (known as punishment) or zero. The goal for the agent is to maximize the expected cumulative discounted rewards by finding an optimal action selection. The learner starts with almost random actions, but by seeking a balance between exploration and exploitation, gradually finds actions that lead to high values of the reward function Sutton & Barto (1998); Kaelbling et al. (1996). Knowing the value of each state, which is the expected long-term reward that can be earned when starting from that state (value function), the agent can choose the best action to take.

Given a system to be controlled. Solutions to this control problem can be to design an RL algorithm to learn the optimal control policy. There is an enormous number of approaches aiming at developing value function based solutions to RL. Temporal difference (TD) is the most common search principle in the space of value functions Sutton (1988). Q-learning is among the TD algorithm which uses the Q-function to estimate the total future reward considering both state and action.

The paper is organized as follows: In next Section we presents the framework of the problem described above and introduces the RL algorithm for the control problem. A common strategy for finding the optimal control policy is the temporal difference approach, for example, which is also presented in next section. Next, a freeway traffic flow model is used to design a freeway controller and the results are presented. Finally, in last Section, conclusions are drawn.

Problem formulation

Consider a nonlinear system, characterized by a state vector $\mathbf{x} = [x_1\ x_2\ \ldots\ x_n]^T$, which can be controlled by an input $\mathbf{u}$, with a fixed discrete time measurement rate, T_s, of its state. System dynamics in discrete-time can be denoted as: $\mathbf{x}_{k+1} = f(\mathbf{x}_k, \mathbf{u}_k)$, with k the discrete time index. The optimal control problem is to control the state of the system from any given initial state $\mathbf{x}_0$ to a desired goal state $\mathbf{x}_K = \mathbf{x}_g$ in an optimal way, where optimality is defined by minimizing the following quadratic cost function:

$$J = \sum_{k=0}^{K-1} \mathbf{e}_{k+1}^T \mathcal{Q} \mathbf{e}_{k+1} + \mathbf{u}_k^T \mathcal{R} \mathbf{u}_k \tag{1}$$

with $\mathcal{Q}$ and $\mathcal{R}$ positive definite matrices of appropriate dimensions and $\mathbf{e}_{k+1} = \mathbf{x}_{k+1} - \mathbf{x}_g$ the error at time $k+1$. The control problem is to find a mapping from states to input $\mathbf{u}_k = \pi(\mathbf{x}_k)$ that, when applied to the system in $\mathbf{x}_0$ results in a sequence of state-action pairs $(\mathbf{u}_0, \mathbf{x}_1), (\mathbf{u}_1, \mathbf{x}_2), \ldots,$ $(\mathbf{u}_{K-1}, \mathbf{x}_g)$ that minimizes the quadratic cost function in (1). The cost function, J, is minimized if the goal is reached in minimum time given the dynamics of the system and restrictions on the size of the input. The importance of speed versus the aggressiveness of the controller is a balance between matrices $\mathcal{Q}$ and $\mathcal{R}$.

In our case, the attempt is to find a control policy for a non-linear system which in general cannot be derived analytically from the system description and the $\mathcal{Q}$ and $\mathcal{R}$ matrices. This limitation holds for many real applications. Note that the RL methods are not limited to the use of quadratic cost functions.

Markov decision processes

The control policy we are attempting to find will be a *reactive policy*, meaning that it will define a mapping from states to actions without the need of storing the states-action pairs of previous time steps. This poses the requirement on the system that it can be described by a state-transition mapping for a quantized state $\mathbf{x}$ and an action (or input) $\mathbf{u}$ in discrete time as follows:

$$\mathbf{x}_{k+1} \sim p(\mathbf{x}_k, \mathbf{u}_k) \tag{2}$$

with p a probability distribution function over the state-action space. In this case, the system is said to be a *markov decision process (MDP)* and the probability distribution function p is said to be the *markov model* of the system. Finding an optimal control policy for an MDP is equivalent to finding the optimal sequence of state-action pairs from any given initial state to a certain goal state, which is a sequential decision problems (SDPs). When the state transitions are stochastic, like in (2), it is a stochastic combinatorial optimization problem. The learning takes place through the agent-environment interaction. The environment is described by an MDP model, which is 5-tuple $(X,\ U,\ P,\ \gamma,\ R)$

- X: A set of discrete states in the system; where $\mathbf{x} \in X$
- U: A set of discrete actions; where $\mathbf{u} \in U$
- $P^{\mathbf{u}}(\mathbf{x}, \mathbf{y})$: Represents the transition probabilities from state $\mathbf{x}$ to state $\mathbf{y}$ when taking action $\mathbf{u}$, Where $\sum_{\mathbf{y}'} P^{\mathbf{u}}(\mathbf{x}, \mathbf{y}') = 1$ and $P^{\mathbf{u}}(\mathbf{x}, \mathbf{y}') \geq 0$
- γ: discount factor, $0 \leq \gamma < 1$, that determines the present value of future rewards
- $R(\mathbf{x}, \pi(\mathbf{x})) \in \mathbb{R}$ is a scalar reward function, where $\pi(\mathbf{x})$ is the policy in state $\mathbf{x}$.

An MDP environment is a process in which the effects of actions depends only upon the current state. The goal of the agent is to find the optimal policy π^*, in which π is defined as a map from state to action and Π is defined as the set of all possible policies. The value of following a policy $\pi \in \Pi$ from state $\mathbf{x}$ is the expected cumulative discounted reward value that can be formally written as:

$$V^{\pi}(\mathbf{x}) = E\left[\sum_{k=0}^{\infty} \gamma^k R\left(\mathbf{x}_k, \pi(\mathbf{x}_k)\right) | \mathbf{x}_0 = \mathbf{x}\right] \tag{3}$$

Having the optimal state-value function, $V^*(\mathbf{x})$, which is defined as $V^*(\mathbf{x}) = \max_{\pi} V^{\pi}(\mathbf{x})$, one can easily find the optimal π, π^*. Finding the value function of a policy is among the convenient ways of finding an optimal policy. Two different solutions to find π^*, one through finding the optimal value function, and the other one through evolution of policy, are presented and discussed in the next two sections.

Temporal difference methods for RL problems

Temporal difference (TD) methods are a class of incremental learning procedures Sutton (1988) that are designed to learn the value function either in on-policy approaches or off-policy methods. Among them, table lookup representation of value function is widely used. For example, the so called Q-learning algorithm stores the return obtained by taking action $\mathbf{u}$ in state $\mathbf{x}$ and

choosing the greedy action w.r.t. the current Q-values. The Q update rule updates the values of state-action pairs according to:

$$Q(\mathbf{x}_k, \mathbf{u}_k) \leftarrow Q(\mathbf{x}_k, \mathbf{u}_k) + \alpha[R(\mathbf{x}_k, \mathbf{u}_k) + \gamma \max_{\mathbf{u}_{k+1}} Q(\mathbf{x}_{k+1}, \mathbf{u}_{k+1}) - Q(\mathbf{x}_k, \mathbf{u}_k)]$$

where α is the learning rate and γ is the discount factor. In the so-called SARSA algorithm, the agent learns the Q-values associated with taking the policy it follows, π, which might be different from the greedy action selection.

$$Q(\mathbf{x}_k, \mathbf{u}_k) \leftarrow Q(\mathbf{x}_k, \mathbf{u}_k) + \alpha[R(\mathbf{x}_k, \mathbf{u}_k) + \gamma Q(\mathbf{x}_{k+1}, \pi(\mathbf{x}_{k+1})) - Q(\mathbf{x}_k, \mathbf{u}_k)]$$

In both algorithms, the value-function parameters are estimated using temporal difference learning, meaning that the temporal difference between the two successive evaluations of the value function is used to update the current value.

Highway traffic flow model

According to application area, the traffic flow models are classified into deterministic vs. stochastic, continuous vs. discrete and microscopic vs. macroscopic representation. METANET Messmer & Papageorgiou (1990) is a deterministic, time discrete-spatially discrete macroscopic modeling tool that results in a good trade-off between efficiency and accuracy. Instead of interacting with the real system, trials are made by interaction with simulation model because of its several advantages including cheaper and faster learning. Details on METANET can be found in Messmer & Papageorgiou (1990). In particular, here we use and extended version of the METANET Hegyi (2004).

Q-learning based density control (agent description)

A freeway control problem of ramp metering can be formulated as Markov Decision Process (MDP) and solved using reinforcement learning algorithms. There are three basic elements components in RL, i.e. state space, action space and reward function. These elements are defined as follows:

States

The state of the network is represented by two variables namely the Density on the downstream of the diversion point and Current metering rate. The first variable is normalized with respect to jam density. The density range of (.2, .4] is discretized to 11 equispaced grid points. The ramp metering rate is discretized into 11 equal parts, ranging from 0 to 1. This state representation is minimal. When the downstream flow is lower than the flow capacity, in sub-optimal case, the agent can select an appropriate action to increase the flow.

Actions

In order to optimize the traffic situation, the agent may select a suitable action, according to its current state. Three actions were considered. The design of actions is show in Table 1. The final ramp metering rate is a finite number of distinct values in [0,1].

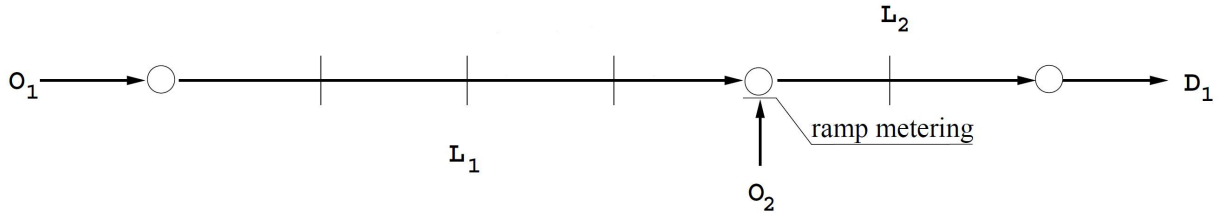

Figure 1: A simple network with an on-ramp Hegyi (2004). The objective of the controller is to maximize the networks throughput by ramp-metering.

The ramp metering will be calculated by the following:

$$r_o\left(k+1\right) = \begin{cases} 0 & \text{if } r_o(k) + \Delta r_o < 0, \\ r_o(k) + \Delta r_o & \text{if } 0 \leq r_o(k) + \Delta r_o \leq 1 \\ 1 & \text{if } r_o(k) + \Delta r > 1 \end{cases} \tag{4}$$

Rewards function

The reward can be either positive or negative, in accordance with the outcome, based on whether a benefit or penalty is accrued. Since the usual control goal is outflow maximization, here we take the reward function as follows:

$$R = \begin{cases} 0 & \text{if } q < .925\ q_{max}, \\ (K-k) * q(k) & \text{if } q \geq .925\ q_{max}, \end{cases} \tag{5}$$

Numerical results

The case study network consists of mainline and an on-ramp. See Figure 1. So basically there are two origins (O_1, the main origin, and O_2), two freeway links (L_1 and L_2), and one destination (D_1). The length of L_1 is 4 km consisting of four segments of 1 km each. Link L_2 has two segments with length of 1 km each, and ends in D_1 with unrestricted outflow. For here, we assume that there is no restriction on the queue length at O_2. The simulation is programmed and carried out using MATLAB. Full details of highway configuration can be found in Hegyi (2004). The problem is that of determining optimal ramp metering control over a time horizon of about 2.5h. Traffic demands of the network is shown in Figure 2.

Table 1: Action set

Action No.	*1*	*2*	*3*
Δr_o	-0.1	0.0	0.1

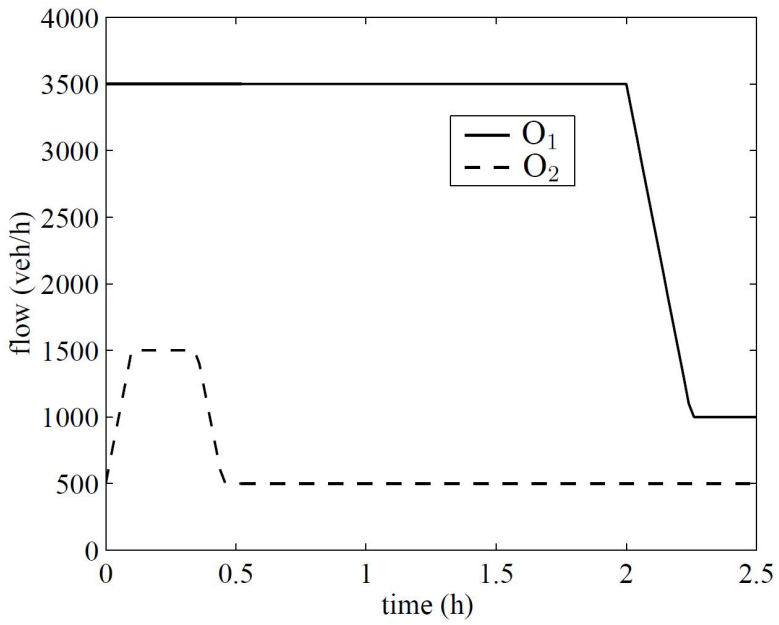

Figure 2: Demand profile over time horizon of about 2.5h.

The ramp metering is considered to be active only if the flow on the downstream of the diversion point is higher than 0.8. The networks characteristics remain the same for all simulation cases. Table 2 displays the utilized parameter values in Q-learning.

Figure 3 presents the simulation results after 1000 iterations. The traffic flow throughput volume is maintained in capacity during the high demands. The reward the agent obtained during one single run is presented in Figure 3(e). To see if the agent can cope with a communication failure (that prevents the agent in communicating with the measures for some time) or any other possible failures, in Figure 4 the ramp metering rate is set to 0 for about 5 minuets. In Figure 5, for the same period, the metering rate is set to 1. The results confirm the robustness of the solution to this type of failures.

Conclusion and future work

The algorithm presented here guarantees a good performance in terms of keeping the flow relatively equal to the capacity. Also the robustness of the controller in terms of possible com-

Table 2: Q-learning utilized parameter values. α is learning rate, γ is discount factor and ε is the parameter value in ε-greedy action selection.

Parameter	α	γ	ε
Value	0.2	0.95	0.2

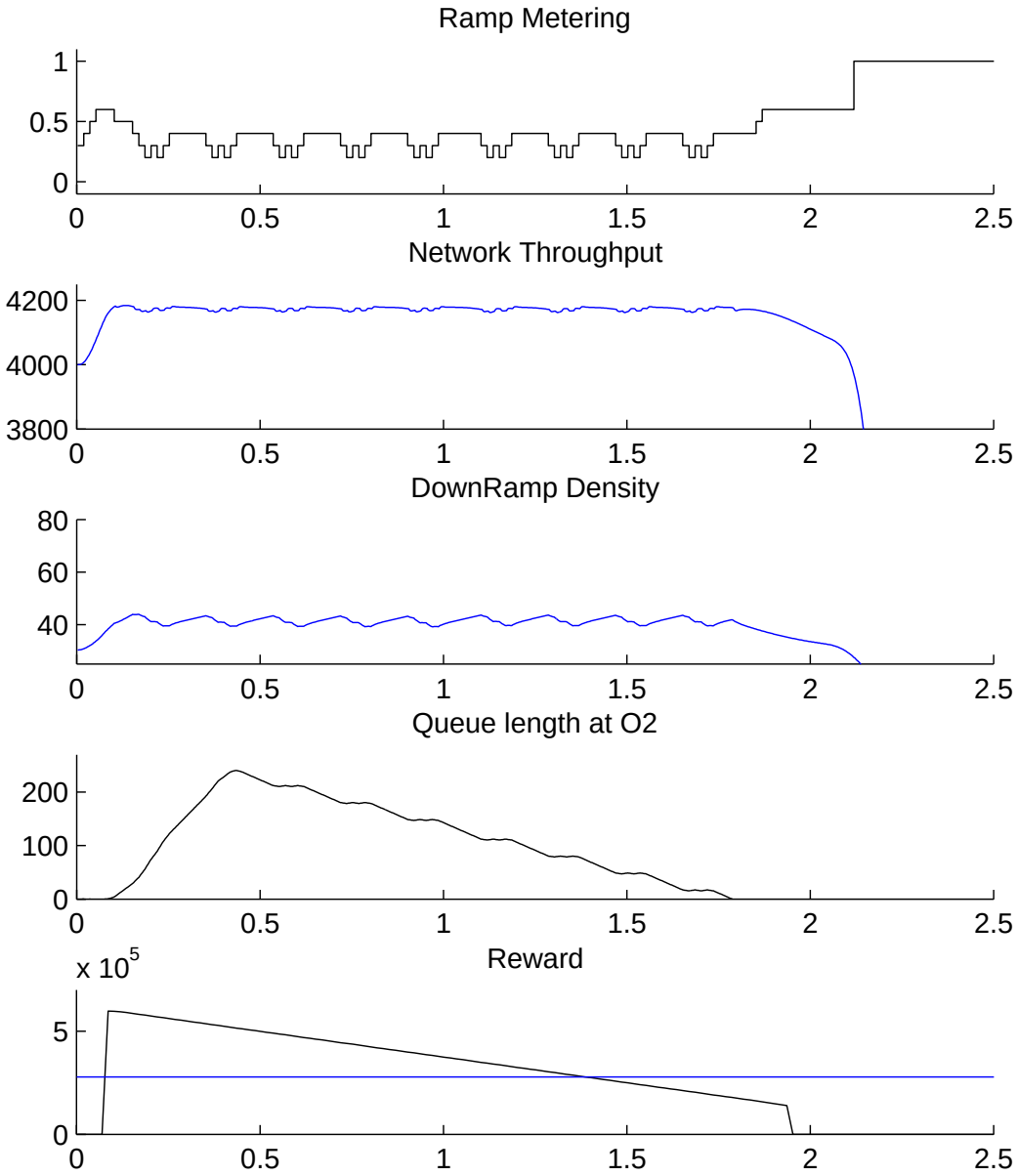

Figure 3: (a) Applied ramp metering control, (b) Resulted flow throughput, (c) Density at fist segment of L_2, (d) Queue length at O_2, (e) Obtained reward values

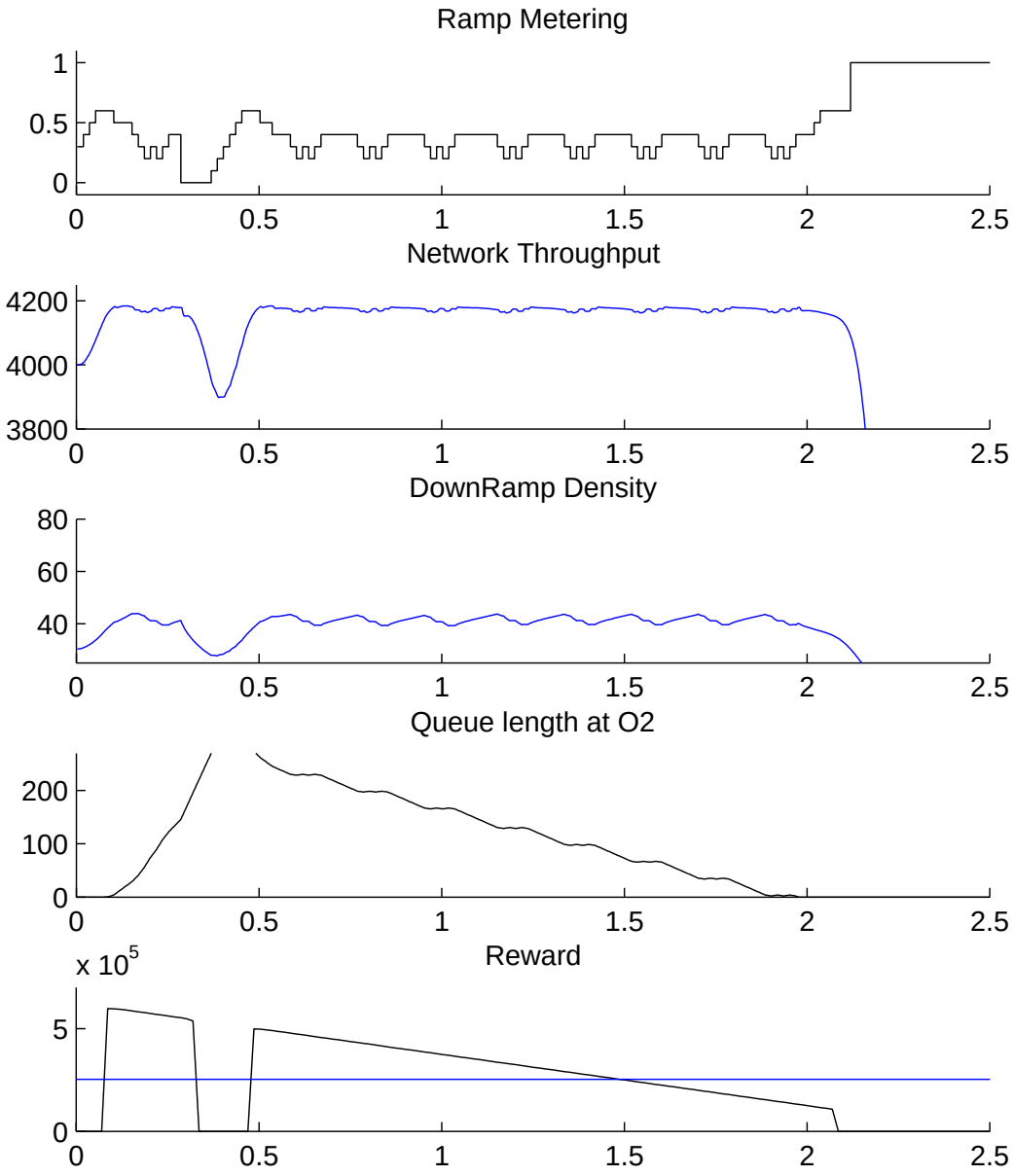

Figure 4: Communication failure over a time horizon of about 5 minutes, 17 minutes after the starting time. (a) Applied ramp metering control, (b) Resulted flow throughput, (c) Density at fist segment of L_2, (d) Queue length at O_2, (e) Obtained reward values

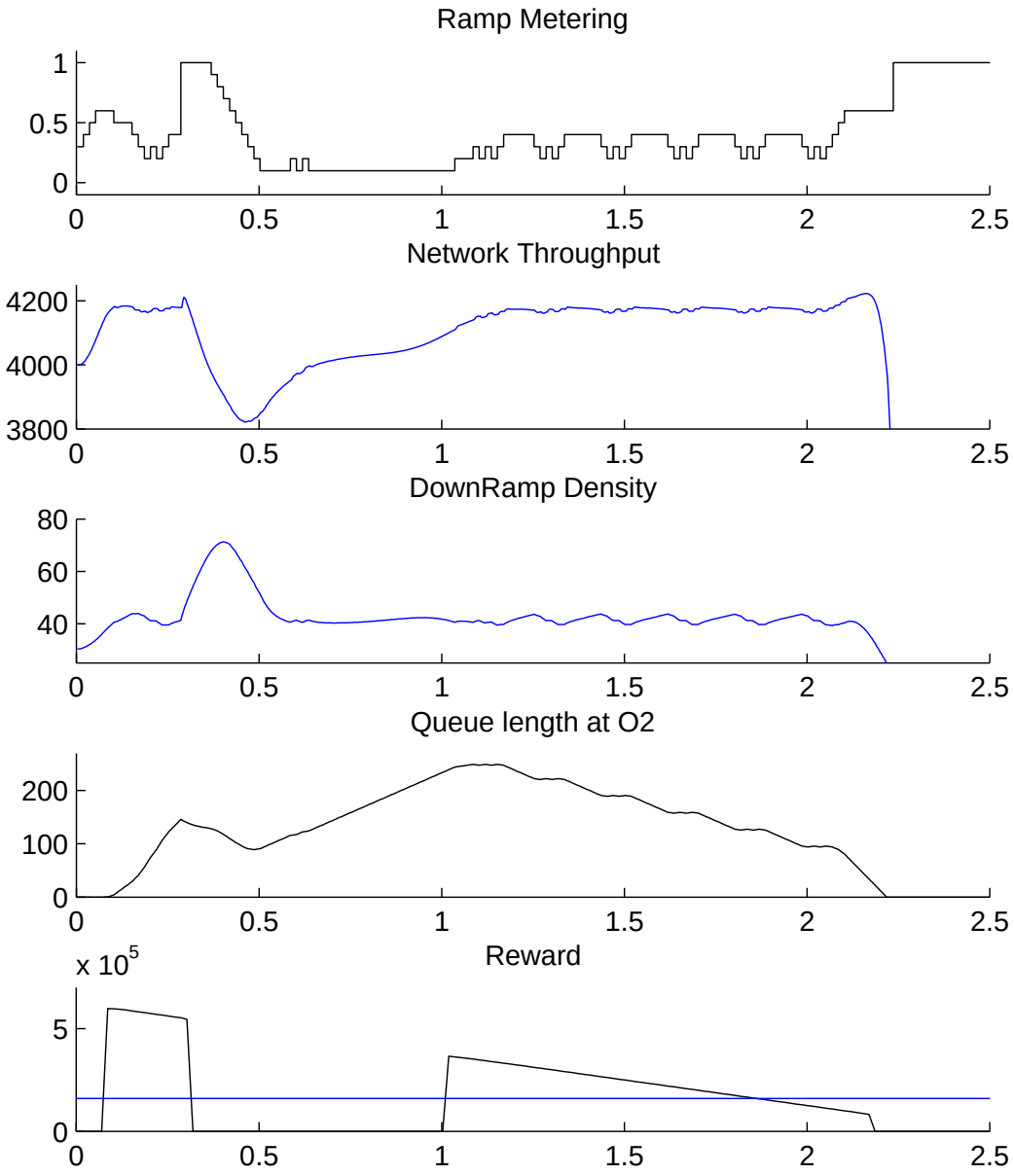

Figure 5: Communication failure over a time horizon of about 5 minutes, 17 minutes after the starting time. (a) Applied ramp metering control, (b) Resulted flow throughput, (c) Density at fist segment of L_2, (d) Queue length at O_2, (e) Obtained reward values

munication failure is investigated. The reward used in the closed-loop setting (e.g. giving only a non-nul reward in the vicinity of the target) can have many different formulations. Theoretically, it is possible to learn a superior controller than the one we archived here by controller with a more informative reward function. An interesting research could be to explore how the formulation of the reward function influence the results.

An interesting setting of ramp metering control problem is to pose input constraints. In practice, when traffic flow exceeds capacity, the queuing is inevitable. In diverse situations, the ramp metering is constrained by the length of the queue on the on ramp. It is necessary to extend the Q-learning based density control to the scenario where the maximum on ramp queue length is bounded by certain upper limit in order to prevent spill-back to a surface street intersection.

REFERENCES

Bellemans, T., B. D. Schutter, B. D. Moor (2003) Anticipative model predictive control for ramp metering in freeway networks, in: *Proceedings of the 2003 American Control Conference*, p. 40774082.

Ernst, D., M. Glavic, F. Capitanescu, L. Wehenkel (2009) Reinforcement learning versus model predictive control: a comparison on a power system problem, *IEEE Transactions on Systems, Man, and Cybernetics - Part B: Cybernetics*, 39(2), pp. 517 – 529.

Hegyi, A. (2004) *Model Predictive Control for Integrating Traffic Control Measures*, Ph.D. thesis, Delft University of Technology, Delft Center for Systems and Control.

Hegyi, A., B. D. Schutter, H. Hellendoorn (2005) Model predictive control for optimal coordination of ramp metering and variable speed limits, *Transportation Research C*, 13(3), p. 185209.

Kaelbling, L., M. Littman, A. Moore (1996) Reinforcement learning: A survey, *Journal of Artificial Intelligence Research*, 4, p. 237285.

Maciejowski, J. (2002) *Predictive Control with Constraints*, Prentice Hal, Harlow, England.

Messmer, A., M. Papageorgiou (1990) Metanet: A macroscopic simulation program for motorway networks, *Traffic Engineering and Control*, 31(8).

Sutton, R. (1988) Learning to predict by the methods of temporal differences, *Machine Learning*, 3, pp. 9–44.

Sutton, R., A. Barto (1998) *Introduction to Reinforcement Learning*, MIT Press.