

BALANCING THE WAITING TIMES IN A SIMPLE TRAFFIC INTERSECTION MODEL¹

Michal Kutil* Zdeněk Hanzálek* Anton Cervin**

* *CAK DCE FEE, Czech Technical University in Prague,
Karlovo nám. 13, 121 35 Prague 2, Czech Republic
{kutilm,hanzalek}@fel.cvut.cz,*

** *Department of Automatic Control, Lund University,
Box 118, SE-221 00 Lund, Sweden
anton@control.lth.se*

Abstract: We propose a novel dynamical model of a simple traffic intersection, where the state variables represent the queue lengths and the mean waiting times in the queues. Including the mean waiting times in the model allows for a more fair traffic control, where the waiting times of the individual vehicles in the various streets of the intersection are taken into account to some degree. The model is linearized and its parameters are estimated using real traffic data measured during one day in Prague. For the balancing of the waiting times, two different controllers are considered: a linear quadratic regulator and a nonlinear model predictive controller. The controllers are evaluated in simulations where real traffic data is used for the incoming flows. *Copyright © 2006 IFAC*

Keywords: traffic queue, intersection control, nonlinear model predictive control

1. INTRODUCTION

In urban traffic control, it is common to decompose the traffic infrastructure into microregions that describe particular streets and intersections. This work focuses on developing a dynamical model of a simple intersection, describing the evolution of the traffic situation by nonlinear difference equations. The objective is to develop controllers for balancing the vehicle waiting times in the different streets of the intersection. For this purpose, we use real traffic data from Prague (Homolová and Nagy, 2005) to tune the intersection model and then develop two controllers: a linear quadratic regulator (LQR) and a nonlinear model predictive controller (NMPC).

The rapid growth of urban traffic requires efficient control methods. The study of *intelligent transportation systems* (ITS) dates back to the 1960s. Since then, a lot of work has been done on road traffic control, freeway traffic control, route guidance, and driver information (see, e.g., (Papageorgiou *et al.*, 2003)). The control of a single intersection (belonging to the class of road traffic control) is usually based on a *fixed-time* strategy or on a *traffic-response* strategy.

In a fixed-time strategy (see, e.g., the TRANSYT tool (Robertson, 1969)), the light control phases (i.e. the duration of green and red light) are scheduled offline. This approach is optimal only in the case of the undersaturated intersections. The light control phases are derived from historical data measured in a given intersection. There are typically several light control phases for each

¹ This work is partially supported by EU/FP6/ARTIST2 and by the Ministry of Education of the Czech Republic under Project 1M0567.

intersection, depending on the given time of the day.

The traffic-response strategies are based on feedback from the current state of the traffic (see, e.g., the SCOTT tool (Hunt *et al.*, 1982)). The *store-and-forward* strategy (Gazis and Potts, 1963; Gazis, 2002) is a traffic-response strategy based on a rigorous mathematical model. The main idea in this model is to introduce a simplification that allows a mathematical description of the traffic flow without the use of discrete variables. This simplifies the description of the system state space and opens up the possibility to use optimization and standard control algorithms. Controllers like NMPC and LQR have been used to control the number of vehicles in the queues—see the traffic-response strategies OPAC (Gartner, 1983), PRO-DYN (Henry *et al.*, 1983), RHODES (Sen and Head, 1997), and TUC (Diakaki *et al.*, 2002).

Our approach builds on the store-and-forward strategy. In our model, we also incorporate the vehicle waiting times, which is a crucial input to the controllers designed in this paper. A similar approach was taken in (Henriksson *et al.*, 2004), where non-linear difference state equations were used to model and control web server traffic.

The remainder of this paper is organized as follows. Section 2 describes the extended queue model. The simple intersection model is given in Section 3 and its control is given in Section 4. Section 5 contains the summary and future work.

2. EXTENDED QUEUE MODEL

Classical traffic control strategies use the single variable n —the *number of vehicles in the queue*, measured in *unit vehicles* [uv]*—*as an input to the control law with the objective to minimize this value. If we want to increase the quality of the traffic control from the driver’s point of view, we can add another objective: the waiting time. The waiting time is the time spent by the vehicle in the queue.

Let us first assume that we are able to track every vehicle and its waiting time in the queue. This will be referred to as the *complete queue model*. The state vector of this model can be written in the form

$$\mathbf{x}_c = (x_1, x_2, \dots, x_i)^T \quad (1)$$

where x_i denotes the number of vehicles that have been waiting in the queue for i time units. The disadvantages of this model are the state equation complexity and the unbounded state vector. In fact, the complexity of the model prohibits the application of standard control techniques.

We next consider an approximate queue model with only two state variables. The first variable,

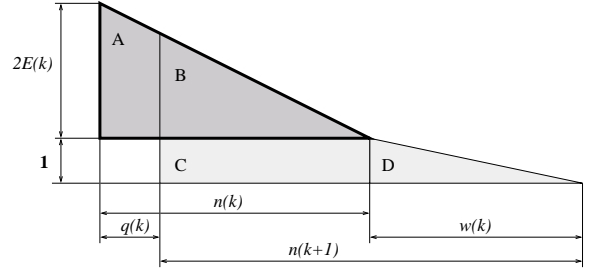


Fig. 1. Extended queue model evolution

n , is the number of vehicles in the queue, while the second variable, E [s], is the *mean value of waiting times*. E is given by S/n , where S is the sum of the waiting times of all vehicles currently in the queue. The state vector of this model is written in the form $\mathbf{x} = (n, E)^T$. This model will be referred to as the *extended queue model*.

2.1 Geometrical Interpretation of the Extended Queue Model

We here derive difference state equations for evolution of the extended queue model. The extended queue model evolution is dependent on the vehicles flows and the length of the time unit. We assume there is an (time-varying) incoming vehicle flow w [$uv \cdot h^{-1}$] and an outgoing flow q [$uv \cdot h^{-1}$] of vehicles leaving the queue.

A geometrical interpretation of the extended queue model is given in Fig. 1. The current state at time k is given by the bold triangle, where the base represents the queue length n and the height represents the longest waiting time in the queue. The longest waiting time is assumed to be twice the mean waiting time E . The area of the bold triangle represents the sum of waiting times over all vehicles: $S = E \cdot n$.

Next, we consider the evolution of the state from time k to $k+1$. The incoming flow during this time interval is assumed to be $w(k)$, while the outgoing flow is $q(k)$. Studying Fig. 1, we have the following geometrical interpretation of the state evolution:

- (1) The outgoing flow $q(k)$ corresponds to the removal of the polygon A from the main triangle. The remaining vehicles are thus given by the triangle B.
- (2) All vehicles staying in the queue increase their waiting time by 1 unit. This corresponds to the addition of the rectangle C.
- (3) The incoming flow $w(k)$ is represented by the addition of the triangle D.

The new area (B+C+D) is equivalent to $S(k+1)$, i.e., the sum of waiting times over all vehicles at time $k+1$:

$$S(k+1) = \underbrace{\frac{E(k)(n(k)-q(k))^2}{n(k)}}_B + \underbrace{n(k)-q(k)}_C + \underbrace{\frac{w(k)}{2}}_D \quad (2)$$

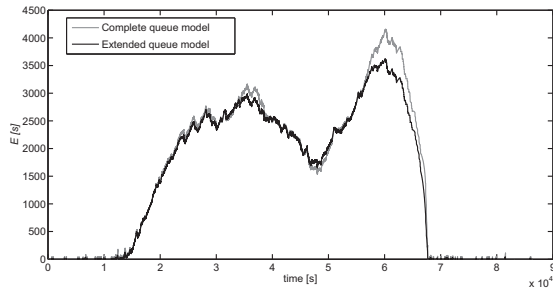


Fig. 2. Queue model evaluation

Finally, using the fact $E(k) = S(k)/n(k)$, we arrive at the following discrete-time state equations:

$$n(k+1) = n(k) - q(k) + w(k) \quad (3)$$

$$E(k+1) = \frac{\frac{E(k)(n(k)-q(k))^2}{n(k)} + n(k) - q(k) + \frac{w(k)}{2}}{n(k) - q(k) + w(k)} \quad (4)$$

These equations are valid only for $n(k) > 0$ and $n(k) > q(k) - w(k)$. This means that there must be some vehicles in the queue, otherwise $E(k+1)$ is equal to 0.

2.2 Extended Queue Model Evaluation

To evaluate the extended queue model, we compared its ability to predict the mean waiting times to that of the complete queue model. (While the complete model has a complex mathematical description, its behavior can be simulated for a bounded number of vehicles.) As input data to both models we used input traffic flows taken from a real traffic region (Homolová and Nagy, 2005). The result is shown in Fig. 2. It is seen that the extended queue model captures the mean waiting times of the vehicles quite well, justifying its use.

2.3 Extended Queue Model Equilibrium

For the purposes of linearization and further control synthesis, we want to find the equilibrium points, i.e., the points where $\mathbf{x}(k) = \mathbf{x}(k+1)$. The equilibrium points for our model must satisfy the conditions

$$n^\circ = n(k+1) \quad (5)$$

$$E^\circ = E(k+1) \quad (6)$$

Solution of these equations implies

$$q^\circ(k) = w^\circ(k) \quad (7)$$

$$2E^\circ(k) = \frac{n^\circ(k)}{q^\circ(k)} \quad (8)$$

(The circle mark means that the value of a given variable is the value in the equilibrium.) The condition (7) means that the incoming flow w must

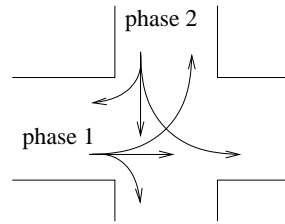


Fig. 3. Simple intersection model

be equal to the outgoing flow q . The condition (8) implies that the mean value of the waiting times is proportional to the queue length and inversely proportional to the vehicle flow. This is the well known Little's law (Little, 1961). In our terminology, the condition says that "the average number of vehicles in a stable queue (over some time interval) is equal to their average incoming flow, multiplied by their average time in the queue."

3. SIMPLE INTERSECTION MODEL

The queue model described above will now be used to construct a *simple intersection model* (see Fig. 3). The intersection consists of two streets (i.e., two queues) and one crossing area (which is a shared resource). The outgoing flow q for each queue is controlled by a semaphore at the intersection.

The simple intersection model is described by

$$\mathbf{x}_M(k+1) = \mathbf{F}(\mathbf{x}_M(k), \mathbf{q}(k), \mathbf{w}(k)), \quad (9)$$

where $\mathbf{x}_M(k) = (\mathbf{x}_1(k), \mathbf{x}_2(k))^T$ contains the state vectors of the two queues. The full intersection state vector is hence given by

$$\mathbf{x}_M(k) = (n_1(k), E_1(k), n_2(k), E_2(k))^T \quad (10)$$

Here, \mathbf{F} is a non-linear function given by Eqs. (3) and (4). The vector $\mathbf{q}(k) = (q_1(k), q_2(k))^T$ represents the outgoing flow for the queues and the vector $\mathbf{w}(k) = (w_1(k), w_2(k))^T$ represents the incoming flow.

3.1 Linear model

A linear model is constructed via linearization of the function \mathbf{F} around an equilibrium point (Section 2.3). The equilibrium point was selected as an average point in the real traffic situation, described by the data in Table 1.

Table 1. Traffic data for the linearization of the intersection model.

Variable	Queue1	Queue2
$q^\circ = w^\circ [uv \cdot h^{-1}]$	100	150
$n^\circ [uv]$	20	50
$E^\circ = \frac{n^\circ}{2q^\circ} [s] (8)$	360	600

The linearized model can be written as

$$\mathbf{x}_M(k+1) = \mathbf{A}\mathbf{x}_M(k) + \mathbf{B}\mathbf{q}(k) + \mathbf{B}_w\mathbf{w}(k) \quad (11)$$

where \mathbf{A} , \mathbf{B} , and \mathbf{B}_w are given by

$$\mathbf{A} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0.05 & 0.99 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0.02 & 0.99 \end{bmatrix}$$

$$\mathbf{B} = \begin{bmatrix} -1 & 0 \\ -17 & 0 \\ 0 & -1 \\ 0 & -11 \end{bmatrix}, \quad \mathbf{B}_w = \begin{bmatrix} 1 & 0 \\ -17 & 0 \\ 0 & 1 \\ 0 & -11 \end{bmatrix}$$

4. CONTROL OF THE SIMPLE INTERSECTION MODEL

The goal of the control is to find an optimal schedule for the traffic lights in the intersection, such that the difference in average waiting times between the two queues is minimized. In this section, two controllers from modern control theory will be designed and compared.

In general, an incoming flow of vehicles arriving at an intersection must be separated into several *phases*. The phase separation, designed by the traffic engineers, determines a direction of vehicles driving through the intersection. A repetitive sequence of phases form a *control period*. The phases have fixed order in the control period and our goal is to find their optimal timing.

The simple intersection model defined above includes the two control phases. Each phase allows vehicles to flow only from one street, see Fig. 3. Our control algorithms consider a constant sum of the phase time intervals, i.e. constant control period T . In this section, the control period is assumed to be 90 seconds. The time when the first phase passes to the second one will be denoted the *switching time* t_{sw} . The switching time can be used to define a control law for the model (9) as follows:

$$\mathbf{q}(k) = \begin{cases} (q_{max1}, 0)^T & \text{if } k \in \langle iT, iT + t_{sw} \rangle, \\ (0, q_{max2})^T & \text{if } k \in \langle iT + t_{sw}, (i+1)T \rangle, \end{cases} \quad (12)$$

Here, q_{maxj} is the maximum feasible outgoing flow from queue j and $i = 0, 1, 2, 3, \dots$ is the index of the control period.

4.1 Linear Quadratic Regulator

In this subsection, a linear quadratic regulator (LQR) (e.g., (Kwakernaak, 1972; Åström and Wittenmark, 1997)) will be used for the intersection control. The objective is to minimize the difference in the waiting times of the vehicles. This

means that a vehicle entering a queue should wait the same time, regardless of which queue it is entering. This can be expressed as minimization of the cost function

$$J = \sum_k (E_1(k) - E_2(k))^2 \quad (13)$$

Using (10), the cost function can be rewritten as

$$J = \sum_k \mathbf{x}_M(k)^T \mathbf{Q} \mathbf{x}_M(k) \quad (14)$$

where

$$\mathbf{Q} = \begin{bmatrix} 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & -1 \\ 0 & 0 & 0 & 0 \\ 0 & -1 & 0 & 1 \end{bmatrix}.$$

Assuming a control law in the form

$$\mathbf{q}'(k) = (q'_1(k), q'_2(k))^T = \kappa \mathbf{x}_M(k)$$

and solving the LQR Riccati equation gives the optimal feedback gain

$$\kappa = \begin{bmatrix} -0.001 & -0.020 & 0.000 & -0.012 \\ 0.001 & 0.016 & -0.001 & -0.062 \end{bmatrix}$$

This control law produces a potentially unbounded result $\mathbf{q}'(k)$, which cannot be directly applied to the intersection traffic control. Instead, from this result we compute the switching time t_{sw} as

$$t_{sw}(k) = T \frac{q'_1(k)}{q'_1(k) + q'_2(k)} \quad (15)$$

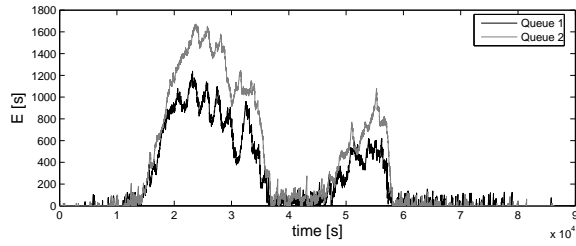
The final control law $\mathbf{q}(k)$ is obtained by combining this expression and Eq. (12). The control law is computed at the start of the control period and is held for the whole control period.

The LQR was applied to the simple intersection model control (9). The simulated response to real input data during one day (i.e. 86400 seconds) is shown in Fig. 4(a). The resulting average waiting times in the two queues are significantly different, the error caused by the linearization of the model.

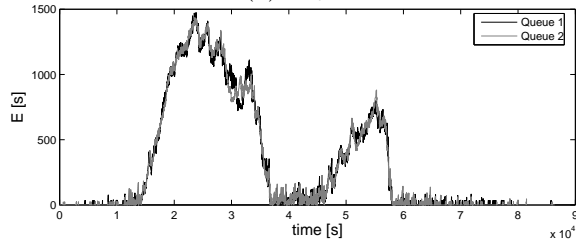
4.2 Non-Linear Model Predictive Controller

Next, we consider controlling the waiting times in the simple intersection model using a non-linear model predictive controller (NMPC) (Findeisen and Allgöwer, 2002; Magni *et al.*, 2003). The same cost function (13) was used as an objective function. For the NMPC algorithm we must select a control horizon and a prediction horizon. The prediction horizon is a time interval that the controller uses for simulating the nonlinear model. The control horizon is a time interval over which the controller optimizes the control signal. In our case, both horizons were set to the 90 seconds, which is equal to the control period T .

For convex problems, the NMPC can find an optimal switching time t_{sw} by convex optimization



(a) LQR



(b) NMPC

Fig. 4. Mean waiting times

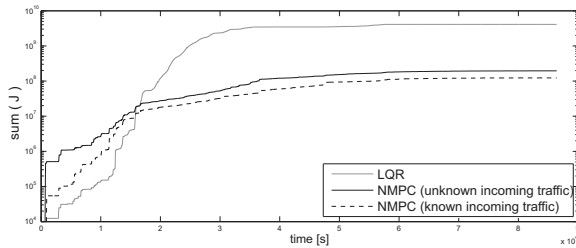
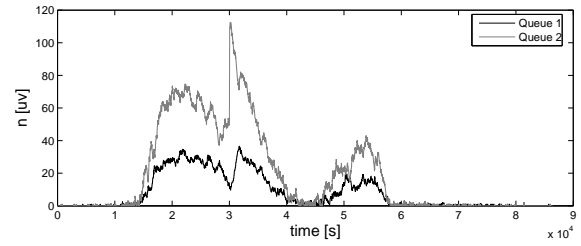


Fig. 5. Evolution of accumulated objective function

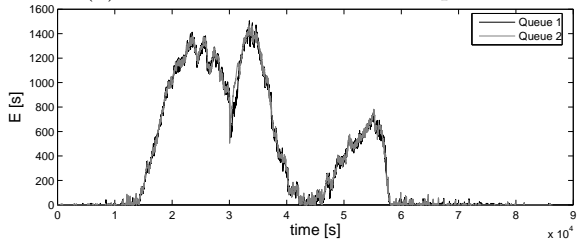
(Boyd and Vandenberghe, 2004). Our optimization problem is not convex, however. Instead, we find the optimal t_{sw} by enumerating all $t_{sw} \in (0, T)$ and simulating the response. The simulated intersection model response when applying the NMPC control law is shown on the Fig. 4(b).

NMPC allows tuning of the control law to be modified in a number of ways. For example, we can extend the controller by taking into account future incoming flow. In practice, we can measure this traffic in a previous, neighboring intersection (as shown by (Lei and Ozguner, 2001)) and forward this information to the next intersection controller. In this way, the predictive controller can prepare a much better control action. Trying this approach on the simple intersection model, adding feedforward traffic information to the NMPC is able to reduce the cost function J by about 37%. The accumulative value of the cost function J for different controllers is depicted in Fig. 5. We can see that the NMPC yields much better results than LQR, and that feedforward from the incoming traffic improves the result even further.

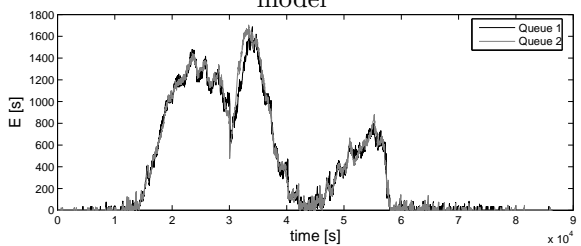
To evaluate the sensitivity of the NMPC to the incoming flow, the following experiment was performed. In addition to the original incoming flow, Queue 2 was subjected to one additional vehicle per second from time 30000 to time 30100.



(a) Number of vehicles in the queue



(b) Mean waiting times in the extended queue model



(c) Mean waiting times in the complete queue model

Fig. 6. NMPC with extra vehicles at $t = 3 \cdot 10^4$

Fig. 6(a) shows that the increase in the number of vehicles in Queue 1 is partially compensated by the NMPC, which leads to an increase in the number of vehicles in Queue 2. From the principle (see Equation 3) the number of vehicles in Fig. 6(a) holds for both models. The mean value of the waiting times in the extended queue model (shown in Fig. 6(b)) is used to calculate the switching time t_{sw} by the NMPC. The same t_{sw} is applied to the complete queue model (see Fig. 6(c)) showing that E in both queues is quite well balanced.

Table 2 shows the complexity of the NMPC calculations in terms of the number iterations for the one-day experiment. The left half of the table shows the complexity results for a prediction horizon of 90 seconds. The first row in the first column refers to the complexity of experiments reported up to now. In general, the control performance can be increased by prolongation of the predictive horizon. In the right half of the table, the complexity results for a predictive horizon of 180 seconds is shown. In all cases, the time complexity is negligible with respect to the control period. Nevertheless, we propose two simple approaches for reducing the problem complexity. First, the t_{sw} does not need to be an integer variable (as assumed in the first row in Table 2), but can be assumed to achieve a value divisible by 2 (the second row) or by 5 (the third row). Second, for

Table 2. NMPC complexity in number of iterations

	predictive horizon 90s			predictive horizon 180s		
	$\delta = 90$	$\delta = 30$	$\delta = 5$	$\delta = 90$	$\delta = 30$	$\delta = 5$
	$\mu = 0$	$\mu = 10$	$\mu = 10$	$\mu = 0$	$\mu = 10$	$\mu = 10$
1	87451	41328	9905	7958041	1980255	110621
2	45167	21927	5995	2122849	548138	39316
5	19220	10055	3587	384400	113237	13724

practical reasons, $t_{sw}(k+1)$ does not need to vary from 0 to 90 (as assumed in the columns with $\delta = 90$, $\mu = 0$), but could be allowed to vary only from $\max\{t_{sw}(k) - \delta, \mu\}$ to $\min\{t_{sw}(k) + \delta, 90 - \mu\}$ where δ stands for maximal allowed difference of t_{sw} and μ defines a minimal duration of the phase. Both approaches lead to a significant reduction of the search space for the NMPC.

5. SUMMARY AND FUTURE WORK

In this paper, a new model for traffic queues has been presented. The extended queue model is described by the number of vehicles in the queue and the mean value of waiting time. The model is based on non-linear difference state equations. We have shown that the equilibrium point of the nonlinear model conforms with the Little's law.

Further, we have used the extended queue model to derive the parameters of the controllers for a simple intersection model with two queues. Two controllers were applied to the intersection model. First, we designed a linear quadratic regulator based on a linearization of the state equations around an equilibrium point. Second, we proposed the use of a nonlinear model predictive controller. The advantages and disadvantages of the controllers were discussed, and their performance was evaluated in simulations using real traffic data.

Current work aims at incorporating several intersections into a traffic microregion model. As future work we would like to include additional practical constraints to the problem (e.g., supervisory systems performing high-level optimization on the model).

In order to model the traffic with higher precision (i.e. incorporating logarithmic stream model capturing the output flow as non-monotonic function of the car density) we are developing a model based on continuous Petri Nets. In the end, we want to compare both models and evaluate the resulting control performance.

REFERENCES

Åström, Karl Johan and Björn Wittenmark (1997). *Computer-Controlled Systems: Theory and Design*. Prentice Hall.

- Boyd, Stephen and Lieven Vandenberghe (2004). *Convex Optimization*. Cambridge University Press. New York, NY, USA.
- Diakaki, C., M. Papageorgiou and K. Aboudolas (2002). A multivariable regulator approach to traffic-responsive network wide signal control. *Control Engineering Practice* **10**(2), 183–195.
- Findeisen, Rolf and Frank Allgöwer (2002). An Introduction to Nonlinear Model Predictive Control. In: *21st Benelux Meeting on Systems and Control*.
- Gartner, N. H. (1983). Opac: A demand-responsive strategy for traffic signal control. *Transportation Research Record* **906**, 75–81.
- Gazis, D. and R. Potts (1963). The oversaturated intersection. In: *Proc. 2nd Int. Symp. Traffic Theory*. pp. 221–237.
- Gazis, Denos C. (2002). *Traffic theory*. Kluwer.
- Henriksson, Dan, Ying Lu and Tarek F. Abdelzaher (2004). Improved Prediction for Web Server Delay Control. In: *ECRTS*. IEEE Computer Society. pp. 61–68.
- Henry, J. J., J. L. Farges and J. Tuffal (1983). The prodyn real time traffic algorithm. In: *4th IFAC-IFIP-IFORS Conference on Control in Transportation System*. BadenBaden, Germany.
- Homolová, Jitka and Ivan Nagy (2005). Traffic Model of a Microregion. In: *IFAC World Congress*.
- Hunt, P. B., D. I. Robertson and R. D. Bretherton (1982). The SCOOT on-line traffic signal optimization technique. *Traffic Eng. Control* **23**, 190–192.
- Kwakernaak, Huibert (1972). *Linear Optimal Control Systems*. John Wiley & Sons, Inc.. New York, NY, USA.
- Lei, Jia and Umit Ozguner (2001). Decentralized hybrid intersection control. In: *Proceedings of the 40th IEEE Conference on Decision and Control*. Vol. 2. pp. 1237–1242.
- Little, J. D. C. (1961). A Proof of the Queueing Formula $L = \lambda W$. *Operations Research* **9**, 383–387.
- Magni, L., G. De Nicolao, R. Scattolini and F. Allgöwer (2003). Robust model predictive control for nonlinear discrete-time systems.. *Int. J. Robust Nonlinear Control* **13**(3-4), 229–246.
- Papageorgiou, M., Ch. Diakaki, V. Dinopoulou, A. Kotsialos and Y. Wang (2003). Review of Road Traffic Control Strategies. *PROCEEDINGS - IEEE* **91**, 2043–2067.
- Robertson, D. (1969). TRANSYT method for area traffic control. *Traffic Eng. Control* **10**, 276.
- Sen, Suvrajeet and Larry K. Head (1997). Controlled optimization of phases at an intersection. *Transportation Science* **31**, 5–17.