APPLICATION OF HYBRID CONTROL METHOD FOR TRAFFIC SYSTEMS

Miroslav Makýš and Štefan Kozák

Slovak University of Technology, Faculty of Informatics and Information Technology, Faculty of Electrical Engineering and Information Technology
Ilkovičova 3, 812 19 Bratislava, Slovak Republic
Tel.: +421 2 60291281 e-mail:makys@fiit.stuba.sk, kozak@stuba.sk

1 DESCRIPTION AND MODEL OF INTERSECTION

Description of intersection

In this study, we will work with intersection with 4 two-way arms. Flows of cars entering the intersection are labelled *A1, B1, B2, C1, C2, D1, D2* and controlled by traffic lights *SA1, SB1, SB2, SC1, SC2, SD1, SD2*, respectively. Each of traffic lights has 3 phases: green, amber and red.

Intersection is shown in figure Fig. 1 from which it is possible to identify which directions are allowed for cars entering the intersection. Cars in *A1, C1* and *D1* flows can only drive straight, cars in *B2, C2* and *D2* flows can drive to the right and from *B1* flow straight and left.

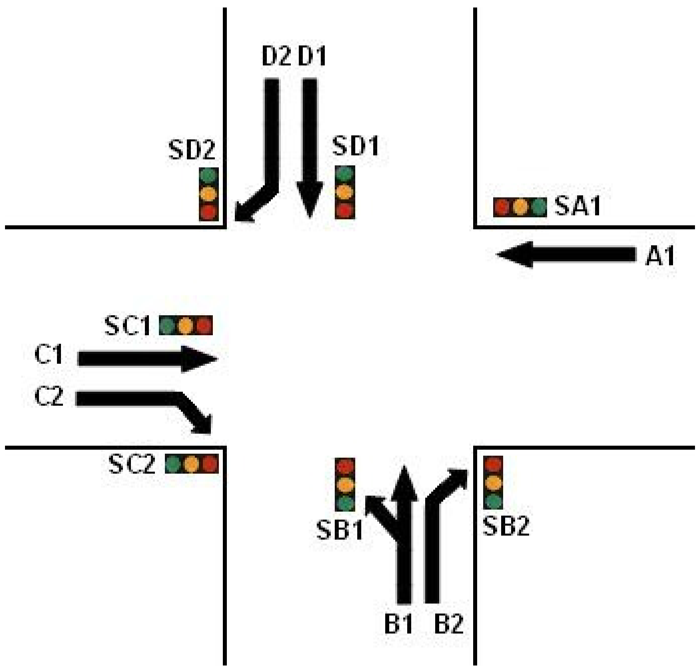

Fig. 1: Examined intersection

The amount of cars coming into the intersection for each of the directions is denoted as λ_i, where $i \in \{A1, B1, B2, C1, C2, D1, D2\}$. Function λ_i is formed by a series of Dirac pulses, and for each of flows is different. One Dirac pulse represents coming of one car. When the traffic light is green or amber, amount of outgoing cars for each of the flows is denoted as μ_i or κ_i, respectively. Functions μ_i and κ_i are also formed by a series of Dirac pulses.

Model of intersection

Let us denote length of waiting car queues as L_i, where $i \in \{A1, B1, B2, C1, C2, D1, D2\}$. When traffic light *Si* is red cars are just coming to intersection when it is green or amber cars are coming and outgoing. Difference of queue length is determined by equation:

$$\frac{dL_i}{dt} = \begin{cases} \lambda_i(t) & \text{if Si ="red"} \\ \lambda_i(t) - \mu_i(t) & \text{if Si ="}green\text{"} \\ \lambda_i(t) - \kappa_i(t) & \text{if Si ="amber"} \end{cases}$$

where $i \in \{A1, B1, B2, C1, C2, D1, D2\}$.

For a realistic model of intersection it is important to define a constraint to the queue length: $L_i \geq 0$. Due to avoid collisions in the intersection it is necessary to impose restrictions on the concurrent green color for determined pairs of traffic lights. These were determined on the basis of intersection specification, the list is in table Tab. 1. Since the *B2* flow is not in conflict with any other flow, it is not necessary to control it and we miss it.

Traffic light 1	**Traffic light 2**
A1	D1
A1	D2
A1	B1
D1	C1
D1	C2
D1	B1
D2	B1
C1	B1

Tab.1: List of flows pairs which can not enter intersection at the same time

Simplified intersection model

Previous intersection model is too complicated for mathematical analysis, therefore, in this section we simplify the model so that it is easier to work with it while still precise enough. It includes following changes:

- lengths of queues are continuous variables,
- comings and outgoings of cars from intersection are represented by constant function,
- amber phase is missing.

Let us denote amount of incoming cars as π_i and amount of outgoing cars as τ_i for each flow where $i \in \{A1, B1, B2, C1, C2, D1, D2\}$.

Difference of queue length is determined by equation:

$$\frac{dL_i}{dt} = \begin{cases} \pi_i(t) & \text{if Si} = "red" \\ \pi_i(t) - \tau_i(t) & \text{if Si} = "green" \end{cases}$$

Intersection model is thus simplified, so that when traffic lights is green, cars are coming and outgoing from intersection in constant rate when traffic lights is red cars are just outgoing in constant rate. Amber phase is omitted.

The model was created by HYSDEL modelling language which was designed for the modelling of hybrid systems. Since the generated model is designed to simulate changes in the length of flows of cars on each intersection arms, state of system is defined as the number of cars waiting in individual flows thus it is vector of length 6. The basic idea is simple: a queue of cars waiting before the intersection is increasing when the light is red and decreasing when the light is green.

Since the intersection is controlled by 6 traffic lights, intersection model will have 6 input control signals one for each traffic light. Traffic light can be green if it does not violate the restrictions in table Tab. 1. For example, if *SC1* light is green also *SC2* and *SD2* lights or *SC2* and *SA1* lights can be green. *SB1* and *SD1* lights have to be red. We need not therefore be subject to each of the traffic lights in particular. With this feature it is possible to reduce the number of control signals, thus simplifying the intersection model and hence the problem of control. Our aim is to determine the minimum number of control signals. Task is therefore to determine minimum normal disjunctive form (MNDF) on the basis of Karnaugh map. Table Tab. 2 shows Karnaugh map for the examining intersection.

MNDF for given map is:

$$(!A_1 \,\&\, !B_1 \,\&\, !D_1) + (!B_1 \,\&\, !D_1 \,\&\, !D_2) + (!A_1 \,\&\, !C_1 \,\&\, !D_1 \,\&\, !D_2) + (!A_1 \,\&\, !B_1 \,\&\, !C_1 \,\&\, !C_2)$$

where ! denotes operator of negation.

If flows from MNDF that to control the intersection we need 4 signals which control the traffic lights following way:

$S1 = !A_1 \,\&\, !B_1 \,\&\, !D_1$ – green light for B_2, C_1, C_2, D_2 flows

$S2 = !B_1 \,\&\, !D_1 \,\&\, !D_2$ – green light for A_1, B_2, C_1, C_2 flows

$S3 = !A_1 \,\&\, !C_1 \,\&\, !D_1 \,\&\, !D_2$ – green light for B_1, B_2, C_2 flows

$S4 = !A_1 \,\&\, !B_1 \,\&\, !C_1 \,\&\, !C_2$ – green light for B_2, D_1, D_2 flows

		D2																
			D2	D2			D2	D2			D2	D2			D2	D2	
				D1	D1	D1	D1					D1	D1	D1	D1		
						C2	C2	C2	C2	C2	C2	C2	C2				
			C1	C1	C1	C1	C1	C1	C1	C1							
		A1	1	0	0	0	0	0	0	1	1	0	0	0	0	0	0	1
B2		A1	1	0	0	0	0	0	0	1	1	0	0	0	0	0	0	1
B2	B1	A1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	B1	A1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	B1		0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	1
B2	B1		0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	1
B2			1	1	0	0	0	0	1	1	1	1	0	0	1	1	1	1
			1	1	0	0	0	0	1	1	1	1	0	0	1	1	1	1

Tab. 2: Karnaugh map for examined intersection

Instead of restrictions listed in the table Tab. 1 we get a new restriction: at most one of the signals can be set to *TRUE*. Because of maximum intersection throughput we can modify this restriction so that it is: just one of the signals must be set to *TRUE*. Let us define vector X as system state and vector U as system input:

$$X = \begin{bmatrix} L_{A1} \\ L_{B1} \\ L_{C1} \\ L_{C2} \\ L_{D1} \\ L_{D2} \end{bmatrix}, \qquad U = \begin{bmatrix} S_1 \\ S_2 \\ S_3 \\ S_4 \end{bmatrix}.$$

Model of system is then defined by:

$$X(k+1) = AX(k) + BU(k) + F$$

where:

$$A = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 \end{bmatrix}, \; B = \begin{bmatrix} -\tau_{A1} & 0 & 0 & 0 \\ 0 & 0 & -\tau_{B1} & 0 \\ -\tau_{C1} & -\tau_{C1} & 0 & 0 \\ -\tau_{C2} & -\tau_{C2} & -\tau_{C2} & 0 \\ 0 & 0 & 0 & -\tau_{D1} \\ -\tau_{D2} & 0 & 0 & -\tau_{D2} \end{bmatrix}, \; F = \begin{bmatrix} \pi_{A1} \\ \pi_{B1} \\ \pi_{C1} \\ \pi_{C2} \\ \pi_{D1} \\ \pi_{D2} \end{bmatrix}$$

2 CONTROL DESIGN

The aim of control is set traffic lights so that throughput of intersection is maximum, while the cars on less busy flows do not wait too long to get to turn. Theoretical maximum throughput is achieved when all the lights are green, which of course due to collisions is not possible. Restrictions are summarized in table Tab. 1.

In this article we'll discuss two approaches to control intersection. The first approach is to achieve the minimum number of cars waiting at an intersection. The second approach sets the green color for the flows where is the biggest sum of waiting cars.

Minimisation of number of cars waiting in crossroads

The aim of this approach is to minimize performance function:

$$J = \sum_{k=1}^{N} \left\| Q_X X(t+k) \right\|_1$$

subject to:

$$X(k+1) = AX(k) + BU(k) + F$$

$$X(t+k) \geq \begin{bmatrix} 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \end{bmatrix}, \text{ for } k \in \{1,2,...,N\}$$

$$\left\| U \right\|_1 = 1$$

where: N is length of prediction horizon,

Q_X is state penalty matrix.

The advantage of this control approach is that the green light lights to flows that most minimize the number of waiting cars. But this is also a disadvantage because the other flows would have green light just after going flows are empty or after approaching defined constraints. But in the next step these flows are not empty (or states are not approaching constraints), so they turn to green, causing continuing switching of lights at an intersection. This behaviour is observable in Fig.2 and Fig.3 graphs. This of course is not desirable because during the time of lights switching no car can pass through the intersection what reduces throughput significantly.

Time responses of the number of cars waiting in individual queues, control input and sum of waiting cars in intersection are depicted on graphs Fig.2-4.

Intersection control was simulated so that after every change of input signal the state of the system was increased what represent number of cars that incomes to queue during lights switching.

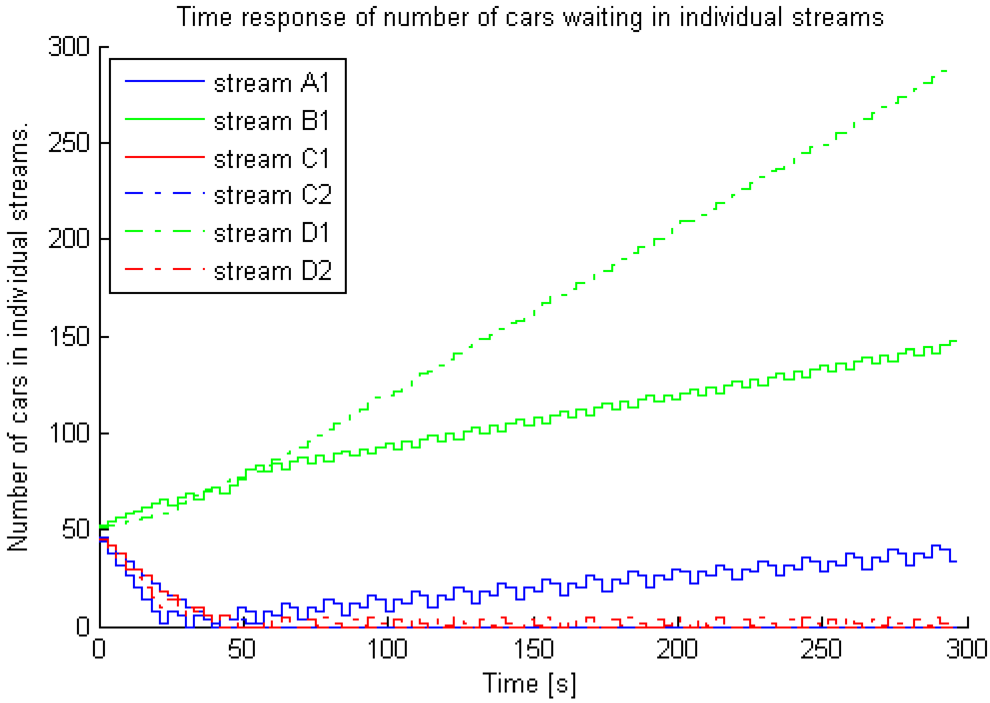

Fig. 2: Time response of number of cars waiting in individual queues

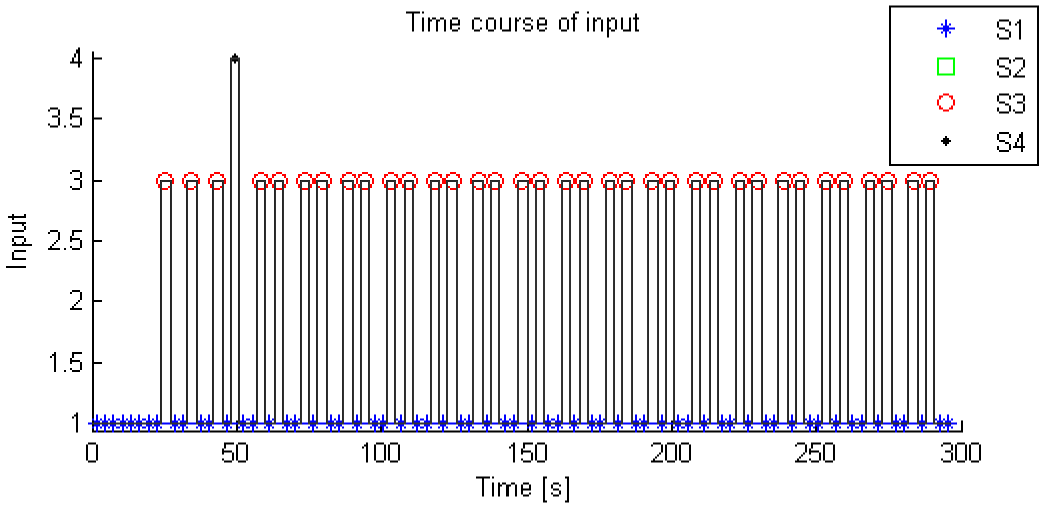

Fig. 3: Input signal *Si* is set to *TRUE* when value of function in graph is *i*.

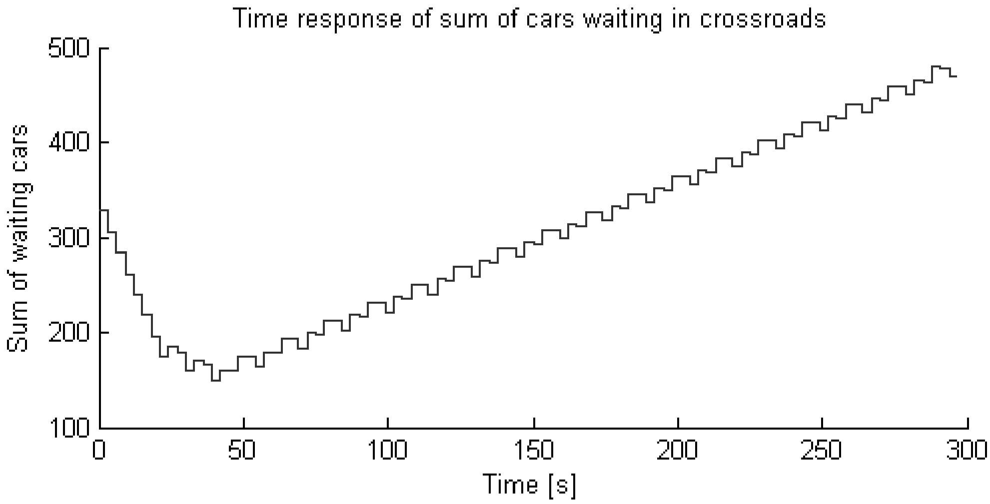

Fig. 4: Time response of total number of cars waiting in intersection

Graphs Fig.2 and Fig.4 show that the number of cars at the intersection is not steady but increasing.

The problem of permanent switching lights can be partially solved by penalizing changes in input signals but we did not achieve good results.

Minimisation of number of cars facing red lights

Aim of second approach for the control of the intersection is to minimize the number of cars facing red lights. To avoid permanent switching of traffic lights performance function penalizes also changes in traffic lights. The aim is to minimize performance function:

$$J = \sum_{k=1}^{N} \left\| Q_x x(k) \right\|_1 - \sum_{k=1}^{N} \left\| Q_L LU \right\|_1 + \sum_{k+1}^{N} \left\| Q_U (U(k-1) - U(k)) \right\|_1$$

subject to:

$$X(k+1) = AX(k) + BU(k) + F$$

$$X(t+k) \geq \begin{bmatrix} 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \end{bmatrix}, \text{ pre } k \in \{1,2,...,N\}$$

$$\left\| U \right\|_1 = 1$$

where: N is length of prediction horizon,

$Q_{X,}$ Q_L, Q_U are penalty matrices,

$$Y = \begin{bmatrix} L_{A1} & 0 & 0 & 0 \\ 0 & 0 & L_{B1} & 0 \\ L_{C1} & L_{C1} & 0 & 0 \\ L_{C2} & L_{C2} & L_{C2} & 0 \\ 0 & 0 & 0 & L_{D1} \\ L_{D2} & 0 & 0 & L_{D2} \end{bmatrix}.$$

Results of this method are shown on Fig. 5-7.

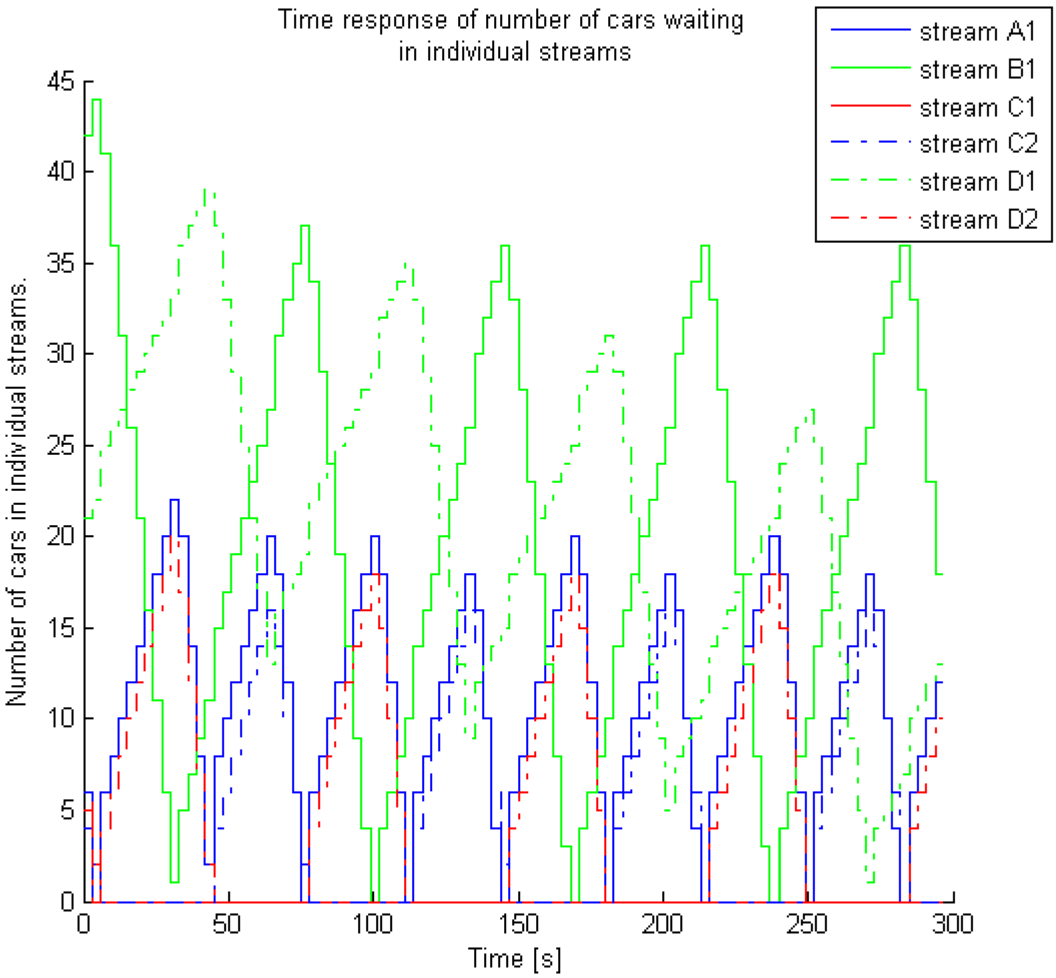

Fig. 5: Time response of number of cars waiting in particular flows

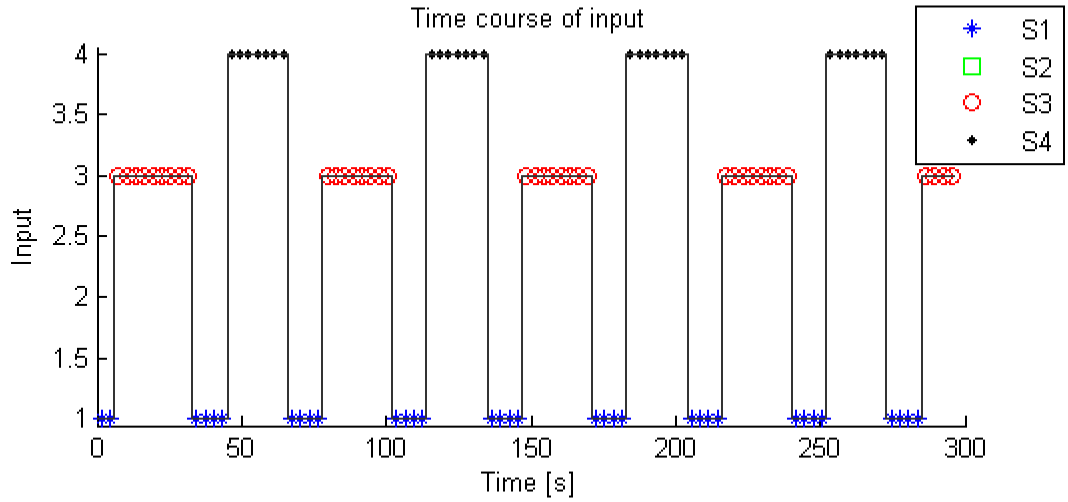

Fig. 6: Input signal *Si* is set to *TRUE* when value of function in graph is *i*.

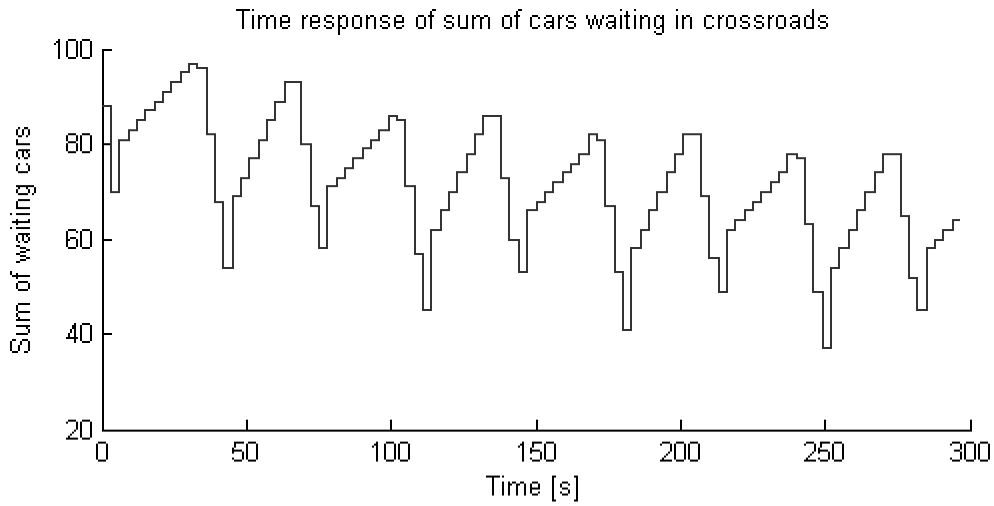

Fig. 7: Time response of total number of cars waiting in flows

Length of prediction horizon

The aim of this section is to show that for used solution when the objective is to minimize the number of cars facing red light there is no need to use longer then 1 step prediction horizon.

Let us suppose that signal *Si* = *TRUE*, other signals are therefore set to *FALSE*. An important condition is that in the current situation and prediction for 1 step is a change in the settings of the signal *Si* to *Sj* in *(n-1)* steps.

Our task is to compare the values of performance function in predicting *n* steps for two cases. The first case is when there is a change of signal in *(n-1)* steps, the second is when signal change is immediate.

Let us set Ψ total number of cars waiting at intersection, Π the total number of cars coming into the intersection (included in the flow) during one sampling period, Ψi number of cars in the intersection which are controller by input signal *Si*, Φ the total number of cars which come to intersection during change of input signal from *Si* to *Sj* (during amber period).

Let us calculate with the first case - a situation where a change occurs after *(n-1)* steps. Number of cars facing red light at time *(t + k)* is:

k=1: $\Psi(t) + \Pi - \Psi i(t+1)$

k=2: $\Psi(t) + 2\Pi - \Psi i(t+2)$

$\vdots$

k=n-1: $\Psi(t) + (n-1)\Pi - \Psi i(t+n-1)$

k=n: $\Psi(t) + n\Pi - \Psi i(t+n-1) - \Psi j'(t+n) + \phi$

Value of performance function is:

$$J_I = n\Psi(t) + \frac{(n+1)n}{2}\Pi - \sum_{k=1}^{n-1}\Psi i(t+k) - \Psi i(t+n-1) - \Psi j'(t+n) + \phi$$

Let us calculate with the second case - a situation where a change occurs immediately. Number of cars facing red light at time *(t + k)* is:

k=1: $(\Psi(t) + \phi) + \Pi - \Psi j(t+1)$

$\vdots$

k=n: $(\Psi(t) + \phi) + n\Pi - \Psi j(t+n)$

Value of performance function is:

$$J_{II} = n\Psi(t) + \frac{(n+1)n}{2}\Pi - \sum_{k=1}^{n-1}\Psi j(t+k) - \Psi j(t+n) + n\phi$$

From the condition if length of prediction horizon is 1 step, then the change of input signal from *Si* to *Sj* occurs after *(n-1)* steps, flows that $\Psi i(t+k) > \Psi j(t+k) + \phi$ for *k = 1, 2, ... n-1*.

Difference of performance functions is:

$$J_I - J_{II} = \sum_{k=1}^{n-1}(\Psi j(t+k) + \phi - \Psi i(t+k)) - \Psi i(t+n-1) - \Psi j'(t+n) - \Psi j(t+n)$$

As

$$\sum_{k=1}^{n-1}(\Psi j(t+k) + \phi - \Psi i(t+k)) < 0$$

$$\Psi j'(t+n) - \Psi j(t+n) > 0$$

$$\Psi i(t+n-1) \geq 0$$

is also $J_I - J_{II} < 0$, so when predicting to *n* steps we get the same results then when predicting to 1 step.

3 CONCLUSION

As a result of the article is proposal of crossroad control based on hybrid predictive control. The main objective for setting traffic lights is number of cars faced to red light. This is a „fair“ approach because we let pass through cars which are in longest queue.

Advantage of proposed control is that we do not need to predict crossroad state lots of steps ahead instead we make just one step prediction.

This project was supported by grant of VG 1/0804/08 (VEGA) project.

REFERENCES

KVASNICA, M., GRIEDER, P., BAOTI, M.: Multi-Parametric Toolbox (MPT). 2004. http://control.ee.ethz.ch/~mpt/

SCHUTTER, B. DE, MOOR, B. DE: Optimal traffic light control for a single intersection. 1998. http://pub.deschutter.info/abs/96_90.html

BEMPORAD, A., MORARI, M.: Control of systems integrating logic, dynamics, and constraints. 1998. http://citeseerx.ist.psu.edu/icons/pdf.gif;jsessionid=857E713B9EE9DAE8B0D8CB3B9EE98FDE