

# Development of road traffic control by using integrated VISSIM-MATLAB simulation environment

Tamás Tettamanti / István Varga

Received 2011-06-17, revised 2011-09-15, accepted 2011-10-12

## Abstract

The paper focuses on the simulation of signal control problems based on VISSIM microscopic simulator. VISSIM is a commonly used software in practice for diverse problems related to road traffic control. Beyond the standard use of VISSIM, advanced options are available. VISSIM COM interface and VISSIM DLL API may be used as well providing freedom to the traffic engineer. By exploiting the programming possibilities efficient simulations may be achieved. As additional possibility MATLAB may be involved in the traffic simulation as well creating an integrated environment with VISSIM. Thus, complex mathematical problems can be calculated online by MATLAB and used for VISSIM simulation.

## Keywords

microscopic traffic simulation · VISSIM · MATLAB · COM · road traffic control · robust control

## Acknowledgement

The work is connected to the scientific program of the "Development of quality-oriented and harmonized R+D+I strategy and functional model at BME" project. The project is also supported by the New Széchenyi Plan (Project ID: TÁMOP-4.2.1/B-09/1/KMR-2010-0002) and by the Hungarian Scientific Research Fund (OTKA) through grant No. CNK 78168 and by János Bolyai Research Scholarship of the Hungarian Academy of Sciences which are gratefully acknowledged.

## Tamás Tettamanti

Department of Control and Transport Automation, Budapest University of Technology and Economics, H-1111, Budapest, Bertalan L. u. 2, Hungary  
e-mail: [tettamanti@mail.bme.hu](mailto:tettamanti@mail.bme.hu)

## István Varga

Computer and Automation Research Institute, Hungarian Academy of Sciences, H-1111, Budapest, Kende u. 13-17, Hungary  
e-mail: [ivarga@sztaki.hu](mailto:ivarga@sztaki.hu)

## 1 Introduction

The continuous development of road traffic control systems is indispensable due to the increasing demands. The control measures can be implemented in three different areas: on freeways, in urban areas or in integrated networks (when the control of freeway and urban roads are managed together). Traffic control strategies may consist of providing basic informations (e.g. congestion warning) or use complex logical algorithms realizing adaptive operations. The actuators are also very heterogeneous, e.g. variable message sign, ramp metering, traffic light, intelligent on-board system, route guidance, etc. In simple static systems (e.g. fixed time ramp metering), the control strategy is realized based on historical measurement data. At the same time, in case of intelligent traffic control the advanced use of informatics is evident. As consequence, when traffic engineers design intelligent traffic control the help of specific computer tools are indispensable. The use of traffic simulators and mathematical optimization software is more and more expected during the development process and the validation phase as well.

Accordingly, the paper investigates advanced simulation possibilities potentially applicable for traffic control design. By focusing on the signal control development, VISSIM COM (Component Object Model) interface and VISSIM API (Application Package Interface) programming are introduced. Furthermore, the integrated VISSIM-MATLAB environment is presented as a state-of-the-art technology for advanced simulation.

As an illustrative example a robust Model Predictive Control (MPC) [5] based strategy is introduced. The control method was efficiently designed and analyzed with the help of the proposed integrated simulation environment.

## 2 Road traffic simulation with VISSIM

Traffic modeling is commonly used in the practice to assist design and validation of newly developed control strategies. Users may choose among many different road traffic simulators. The greater part of them are commercial software. Additionally, open source simulators are also available, developed by universities or research institutes. Each of them has advantages or drawbacks depending on the individual demands of the user.

In the paper, the proposed simulation environment is designated to be designed by using VISSIM [13].

VISSIM is a microscopic simulator based on the individual behavior of the vehicles. The goal of the microscopic modeling approach is the accurate description of the traffic dynamics. Thus, the simulated traffic network may be analyzed in detail. The simulator uses the so-called psycho-physical driver behavior model developed originally by Wiedemann [18]. VISSIM is widely used for diverse problems by traffic engineers in practice as well as by researchers for developments related to road traffic.

VISSIM offers a user friendly graphical interface (GUI) [13] through of which one can design the geometry of any type of road networks and set up simulations in a simple way. However, for several problems the GUI is not satisfying. This is the case, for example, when the user aims to access and manipulate VISSIM objects during the simulation dynamically. For this end, an additional interface [12] is offered based on the COM which is a technology to enable interprocess communication between software [3]. The VISSIM COM interface defines a hierarchical model in which the functions and parameters of the simulator originally provided by the GUI can be manipulated by programming. It can be programmed in any type of languages which is able to handle COM objects (e.g. C++, Visual Basic, Java, etc.). Through VISSIM COM the user is able to manipulate the attributes of most of the internal objects dynamically. The first step of a COM based simulation is to create the COM client. Then, one can realize the parts or even the whole process of the simulation.

As further possibility Dynamic Link Library (DLL) interface programming is also available for specific parts of the simulator. The APIs for this option is provided by the VISSIM software manufacturer written in C++ language. This allows flexibility for the user as one may freely create own developments for the simulations. On the other hand, the simulation run is much faster by using self-developed DLLs for some specific parts compared to the the same simulation controlled by COM interface programming.

VISSIM DLL interface programming is available for driver model, emission model, signal control and toll pricing.

### 3 Traffic signal control with VISSIM

VISSIM GUI provides several possibilities to choose for simulation of traffic controllers. In case of static control, the logic can be easily defined by using the Fixed time module. Fixed time signal controller is available for editing directly through the VISSIM GUI. In addition, one can design phase-based and stage-based signal plans. User may design actuated control with the VAP (Vehicle Actuated Programming) module allowing to create arbitrary logic. The module is programmable by the VAP language or flowchart editor (VISVAP) provided by VISSIM. The option of External signal controller may be chosen as well through of which user-defined signal control can be used. In this case, however, user has to apply the Signal Control API. Hence,

by using the object oriented C++ language more complex logics can be resulted compared to VAP. Beside the previous options VISSIM offers further control modules which are commercial products adapting to the market demands (e.g. SCATS, SCOOT, SIEMENS VA, LISA+).

In the following parts, methods using VISSIM COM and VISSIM API will be introduced focusing on the design of customized control logic. Furthermore, an advanced application possibility is introduced by using MATLAB to assist VISSIM simulation.

#### 3.1 Signal control with VISSIM COM interface programming

To create complex signal control logic one can use VISSIM COM interface exploiting the advantages of the current programming language. By using the attributes of the SignalGroup interface of VISSIM COM, one can realize arbitrary signal control. For example, attribute STATE is able to set any signal state (red, red-amber, green, etc.) on the signal heads at any time.

The following steps formulate a general synthesis for the realization of any adaptive control logic through VISSIM COM interface:

- 1 One generates the overall traffic network through VISSIM GUI (geometry, signal heads, detectors, vehicle inputs, etc.).
- 2 After choosing a programming language which allows COM interface programming, one creates the COM Client.
- 3 By using the attributes and functions of the SignalGroup interface of VISSIM COM, one is able to realize signal control logic.
- 4 The program must be compiled and build as an executable file (EXE) which can be run together with the corresponding VISSIM project file (INP) and initialization file (INI) resulted by the GUI.

If one wishes to design logic through VISSIM COM interface Fixed time controller type should be chosen. In this case, Fixed time module can be used also for adaptive control as at the end of each control interval the states of all signal groups can be redefined based on the user-defined logic. Except the Fixed time module, in case of the other control types (VAP, External, etc.), it makes no sense to manipulate the signal control via VISSIM COM if they are designed and used in a standard way.

#### 3.2 Signal control with VISSIM API programming

Another possibility to create own control logic is the programming of the VISSIM API which provides an advanced application. One may choose the External type controller which allows to connect external control to the simulator through the Signal Control API. By using this method, separate DLL file must be defined for each Signal Controller which is called at each controller time step (1 sec by default) during the simulation. When

the simulator contacts the DLLs the current signalization states and detector data are passed to the corresponding DLLs. Then, by using the acquired data, DLL files calculate the new desired signal states which will be passed back to the simulation. Depending on the settings, either the new signal states are applied immediately, or transition signal states (amber, red-amber) are used first automatically.

Together with the Signal Control API another package, the Signal GUI API interface is also provided. This package can be used for addition of dialog boxes "Edit Parameters" in the Signal control menu (External type) of VISSIM GUI. Through the Signal GUI API the following parameters of the controllers can be edited: name, cycle time, amber, red-amber, minimum green, minimum red and red clearance times. Thus, only the starting times of green and red signs must be defined in the control logic. In case of urban traffic simulation, the possibility to predefine transition times means an important advantage compared to the VISSIM COM based traffic control (Section 3.1) in which these states have to be set between red and green phases again and again by the user's code.

Signal GUI DLL files are called in two cases. On the one hand, when the user selects "Edit Parameters" in the Signal control menu. On the other hand, at the start of each simulation run to load parameters for VISSIM.

The steps below summarize the general realization of external control logic through VISSIM API:

- 1 One generates the overall traffic network through VISSIM GUI.
- 2 By using the attributes and functions of the Signal Control API and Signal GUI API (in C++ language), one is able to realize arbitrary signal control logic with predefined transition times.
- 3 The developed program must be compiled and build as DLL.
- 4 The DLLs must be linked to the project file through the VISSIM GUI.

### 3.3 Advanced traffic simulation in VISSIM-MATLAB environment

It is obvious that the application of VISSIM COM or VISSIM API may efficiently help the user to create customized simulations in case of complex problems. At the same time, it may worth making calculation by other software concerning some specific parts of the simulation. In case of optimization procedures, for example, the user does not necessarily need to know the optimization algorithm in detail to use it right. Thus, by using specific mathematical software, lots of energy and time can be saved compared to the realization by programming. To exploit other software's functionality for VISSIM the main precondition is the online availability and communication during the simulation. Therefore, as further possibility, the use of MATLAB is proposed to assist VISSIM simulation. MATLAB

is a widely used scientific, mathematical software among engineers and researchers. It provides a plethora of functions and allows the implementation of algorithms, or interfacing with programs written in other languages including C, C++, Java, and Fortran. MATLAB can be controlled through COM interface similarly to VISSIM. Therefore, MATLAB may be used as a programmable mathematical subroutine library. By programming the MATLAB Engine API [11], provided by the MATLAB manufacturer, the following functionalities may be used:

- Data interchanges between MATLAB and user's program.
- Call of any MATLAB routines from user's program.

For all these purposes MATLAB can be efficiently used for VISSIM simulation as well. Matlab Engine API is easily programmable either through VISSIM COM interface or VISSIM API. Thus, a complete VISSIM-MATLAB simulation environment (Figure 1) can be realized.

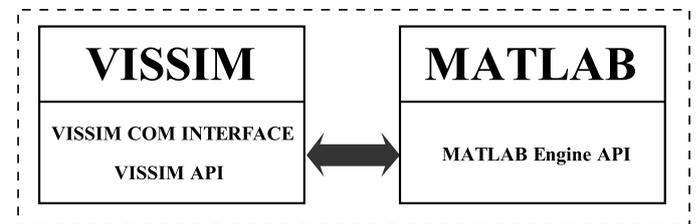


Fig. 1. The complete simulation environment

From a practical point of view the following steps need to be done for using MATLAB Engine with VISSIM by using C++ language:

- 1 VISSIM COM (EXE) or VISSIM API (DLL) project must be created.
- 2 Interface file engine.h (available in MATLAB installation library) has to be added to user's project file.
- 3 Additional MATLAB library files (LIB) must be linked to user's program.

By using the routines of the engine library, VISSIM may be used together with MATLAB. Beside other functions, the engine routines allow to get MATLAB array from the engine, send array to the engine, and execute MATLAB routines.

### 4 Application example: robust traffic control

As a representative example for the use of the integrated VISSIM-MATLAB simulation an advanced road traffic control application is presented. The application was realized and tested by the proposed simulation environment. Our application example presents a robust minimax MPC which is able to handle the model uncertainty problem.

Nowadays, intelligent traffic-responsive strategies are highlighted in urban traffic control. During the past decades, several results have been published in this field providing network-wide

optimal control algorithm. These solutions intend to assure real-time and adaptive control maximizing some performance criterion (e.g. minimize the average travel time) [2, 4, 14–16]. The main disadvantage of these solutions is the incapability of explicitly dealing with model uncertainties. In urban traffic several forms of state disturbances can be present: unexpected traffic fluctuations (demand and exit flows within the link) caused by parking places along the road or non-controlled (and non-measured) junctions in the network (see Figure 2). These distur-

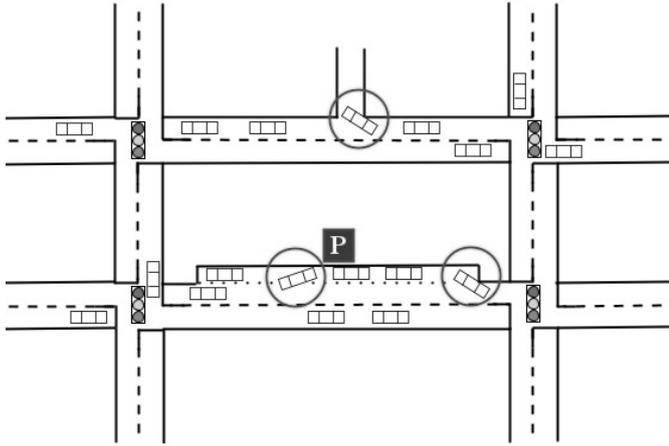


Fig. 2. Uncertainties in urban traffic network

bances can result in strong unmeasured state variations producing significant model uncertainties.

#### 4.1 Modeling and robust control in urban traffic network

A common macroscopic modeling approach for urban road traffic is based on the Store-and-forward dynamic model [6] representing a simple vehicle-conservation equation between signalized and neighboring intersections  $M$  and  $N$  on link  $z$ :

$$x_z(k+1) = x_z(k) + \frac{T}{C} \left[ \sum_{w \in I_M} \alpha_{w,z} S_w \sum_{i \in v_w} u_{M,i}(k) - S_z \sum_{i \in v_z} u_{N,i}(k) + d_z(k) \right]. \quad (1)$$

$x_z(k+1)$  is the number of vehicles expressed in Passenger Car Equivalent (PCE) during  $[kT, (k+1)T]$ , where  $k$  is the discrete time step and  $T$  the sampling time.  $C$  is the cycle time and we assume  $T = C$ .  $I_M$  represents the set of incoming links of junction  $M$ .  $\alpha_{w,z}$  is the turning rate of link  $w$  entering junction  $M$  towards link  $z$ .  $S_w$  and  $S_z$  are the saturation rates representing the capacity of the outflow of the link during its green time. The values of  $\alpha$  and  $S$  are considered known and constant in the paper. Nevertheless, they can be assumed to be time-variant and may be continuously measured or estimated, e.g. by on-line estimation with state observer method [8].  $\sum_{i \in v_w} u_{M,i}$  and  $\sum_{i \in v_z} u_{N,i}$  represent the green times of intersections  $M$  and  $N$ , respectively, during the cycle time.  $v_w$  and  $v_z$  denote the sets of stages where link  $w$  and  $z$  have right of way.  $d_z$  denotes the en-

tering demand appearing at the boundary of the traffic network, expressed as PCE during sample time interval  $T$ .

In order to obtain network dynamics the link equations (1) can be written in matrix form yielding the general linear time-invariant (LTI) state space representation [1]:

$$x(k+1) = Ax(k) + Bu(k) + d(k), \quad (2)$$

where the elements of state vector  $x(k) = [x_1, x_2, \dots, x_n]^T$  represent the number of vehicles of each signalized link. The values of  $x(k)$  are not directly measurable but may be estimated by using Kalman-Filter [17], for example. State matrix  $A$  is considered as constant identity matrix in practice. Control input vector  $u(k)$  contains the green times (in seconds) of all stages. Their numerical values are the results of the corresponding controller at each cycle. The number of control inputs is equal to the number of controlled links. Control matrix  $B$  is composed of the combinations of turning rates and saturation rates.  $B$  is considered as constant with fixed nominal rates. Vector  $d(k)$  contains the entering demand traffic if exist.  $d(k)$  is a time-varying term. In practice, however, it is more reasonable to re-define it in shorter time periods based on continuous real-time measurements.

Traffic model (2) is only valid with state and control input constraints. The maximum number of vehicles (states) is determined by the length of link  $z$  between two junctions. Control input (green time) is subjected to two types of constraints. On the one part,  $u$  is restricted to be in a given time interval. On the other part, the sum of the green times has to be less than the difference of  $T$  and the fixed lost time (resulted from the geometry of junction  $j$ ).

#### 4.2 Uncertainty modeling in urban traffic

A common approach for modeling uncertainties in state space framework is the use of additive uncertainty model [9] which means a simple addition to state equation (2):

$$x(k+1) = Ax(k) + Bu(k) + d(k) + G(k)w(k), \quad (3)$$

$$w(k) \in \mathbb{W}_\infty.$$

Disturbance  $w(k)$  is only known to be bounded, but otherwise unknown. The set of possible disturbances is denoted by  $\mathbb{W}_\infty$  which represents a box-constrained uncertainty model:

$$\mathbb{W}_\infty = \{w : \|w\|_\infty \leq 1\}. \quad (4)$$

Consequently, the maximal value of  $w_z(k)$  may vary between  $-1$  and  $+1$ . Hence, matrix  $G(k)$  must be used for scaling the measure of uncertainties.

It is apparent that (3) is easily applicable for urban traffic problem if we intend to model unmeasurable demand and exit flows. Knowing the bounds of the possible disturbances, matrix  $G(k)$  can be defined. The maximal potential uncertainties ( $\gamma_z$ )

are placed into the diagonal elements of  $G(k)$ .

$$G(k) = \begin{bmatrix} \gamma_1(k) & 0 & \dots & 0 \\ 0 & \gamma_2(k) & & \vdots \\ \vdots & & \ddots & 0 \\ 0 & \dots & 0 & \gamma_n(k) \end{bmatrix}, \quad (5)$$

where  $\gamma_z$  represents number of vehicles. Thus, the bounded additive disturbance on state  $x_z$  is given by product  $\gamma_z w_z(k) \in [-\gamma_z, \gamma_z]$ . In case of the determination of an uncertainty model for traffic network, the first step is to construct time-varying matrix  $G(k)$ . This can be done by the adequate estimation of the disturbances, based on prior observations exclusively. Although  $G(k)$  is time-varying, it does not definitely alter in each time step. Therefore, it is more favorable to predefine uncertainty bounds for larger time intervals. Traffic flows induced by the parking lot of an office building, for example, can be given separately for early morning, late morning, midday hours, etc.

#### 4.3 Minimax approach for robust traffic control

A possible control method to deal with additive uncertainty was published in [9], which is the basis of our traffic control design. To realize robust traffic control strategy the following optimization of the robust performance objective must be solved:

$$\begin{aligned} \min_u \max_w & \sum_{j=0}^{N-1} \{x(k+j|k)^T Q x(k+j|k) + \\ & u(k+j|k)^T R u(k+j|k)\}, \\ \text{subject to} & \quad u(k+j|k) \in \mathbb{U}, \quad \forall w \in \mathbb{W}_\infty, \\ & \quad x(k+j|k) \in \mathbb{X}, \quad \forall w \in \mathbb{W}_\infty, \\ & \quad w(k+j|k) \in \mathbb{W}_\infty. \end{aligned} \quad (6)$$

$x$  is the vector consisting the numbers of vehicles of the controlled links.  $u$  is the vector of green times representing the control inputs.  $Q \succ 0$  and  $R \succ 0$  are symmetric weighting matrices.  $N$  is the prediction horizon.  $\mathbb{U}$  and  $\mathbb{X}$  represent the constraint sets of control input and state considering the possible disturbance set  $\mathbb{W}_\infty$ . The optimization problem aims to find optimal control input which is able to minimize the vehicle queue lengths in the network with respect to the maximal possible uncertainties. Instead of the direct solution of (6), the minimax problem can be replaced with a conservative approximation [7, 9]. (6) can be rewritten as an optimization problem involving linear matrix inequalities (LMI), which is more efficiently solvable by applying SDP algorithm.

#### 4.4 Control strategy design in VISSIM-MATLAB environment

It is obvious that the realization of the SDP optimization based control algorithm, proposed in Section 4.3, would mean a difficult problem in C++ language. Therefore, the use of MATLAB is advised for simulation goals. By using the integrated VISSIM-MATLAB simulation environment, the robust strategy can be efficiently designed and tested.

A fictional test network was created including 4 signalized intersections with 16 measured links (states) and 2 uncontrolled junctions to generate unmeasurable inflow and outflow as uncertainties (represented schematically by Figure 3). The lengths of

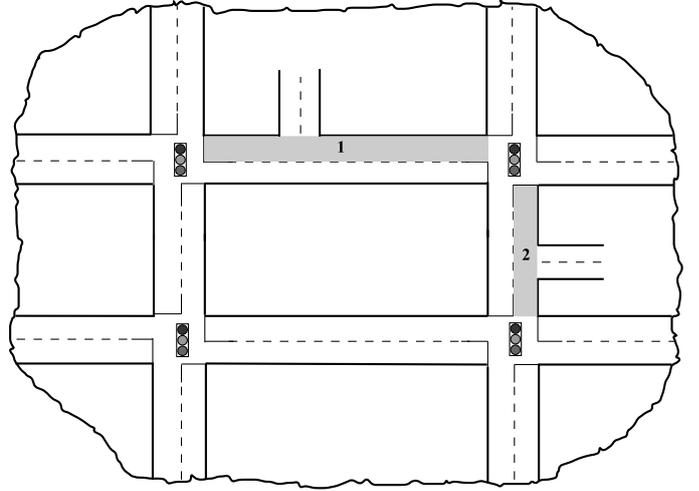


Fig. 3. The test traffic network with two uncertain links

links were defined between 200 and 300 meters. The entering demands ( $d$ ) at the boundary of the network were determined by considering average traffic measurements in District X., Budapest during peak hours in the morning. Nevertheless, the simulation network was fictional. The gray areas in Figure 3 denote the links facing the problem of bounded additive disturbances. The uncertainties in the control scheme were defined based on average historical traffic flows of the disturbed links during the sample time  $T: \bar{n}$ . Three different scenarios were simulated representing maximal potential uncertainties as  $\pm 10\%$ ,  $\pm 20\%$  and  $\pm 30\%$  of  $\bar{n}$ . Thus,  $\gamma_z(k)$  in (5) was given as the proportion of  $\bar{n}_z$  in the different scenarios:

$$\gamma_z(k) = \mu \cdot \bar{n}_z, \quad (7)$$

where  $\mu = 0.1$ ,  $\mu = 0.2$  and  $\mu = 0.3$ . The disturbing vehicle fluctuations generated for the traffic simulation were redetermined cycle by cycle by using the following formula:

$$r_z(k) \cdot \gamma_z(k), \quad (8)$$

where  $r_z(k) \in [0, 1]$  is the random parameter generated by C++ random function to scale  $\gamma_z(k)$ . The dynamics ( $r_z(k) \cdot \gamma_z(k)$ ) and the direction of the disturbing traffic fluctuations were realized through VISSIM COM interface randomly.

We chose the Signal Controller DLL programming for signal control. However, the DLL interface was used only for detector data and signal state settings. The robust control algorithm were realized in MATLAB environment. The connection was established between MATLAB and the DLL by using the MATLAB Engine API (Section 3.3). The SDP optimization problem was solved online by using YALMIP toolbox [10] of MATLAB to provide optimal calculation for green signal splits.

The programmed DLL file provided all detector measurements of the traffic network for MATLAB at each sample time. After the optimization procedure run in MATLAB the optimal green times were sent back to the DLL for signal settings. The control process run once in each cycle while VISSIM simulated continuously. The complete simulation environment is represented by Figure 4.

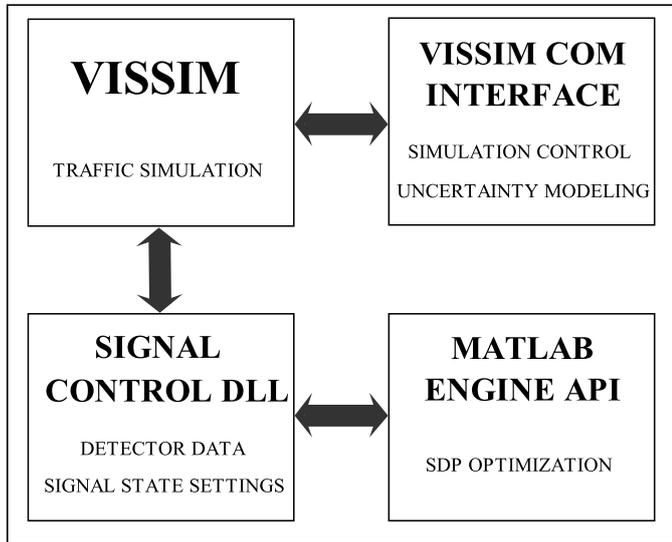


Fig. 4. Simulation environment of the robust traffic control

To measure the effectiveness nominal MPC (NMPC) and robust MPC (RMPC) control strategies were compared in presence of additive uncertainties. All strategies were simulated in all scenarios. In the first case optimization problem (6) was applied without maximization, since the NMPC did not take any uncertainties into consideration. At the same time, strategy RMPC was realized as SDP optimization based on (6). The simulation time was 1800 seconds. The control cycle time was set to  $T = 90$  [sec] in each intersection. We chose control horizon  $N = 5$  for (6).

Four illustrative traffic parameters were selected which are provided by VISSIM for evaluation. The results are displayed in Table 1. Row *Change* represents the variations of traffic parameters achieved by robust control strategy compared to the nominal MPC.

The results show that all traffic parameters changed in the right way, due to the robust strategy. The results are promising compared to the nominal case. Nevertheless, in the case of  $\pm 10\%$  maximal potential uncertainty the measure of amelioration is not significant. This implies that the computational effort is not unambiguously worth if the possible disturbance is considerably low.

By using the integrated simulation environment a complex simulation setup could be realized. The simulation results efficiently helped to improve the control algorithm and provided useful results for final validation.

Tab. 1. Simulation results of the nominal and robust control strategies

Uncertainty	Control strategy	Average delay time per vehicle [sec]	Average number of stops per vehicle	Average speed [km/h]	Average total travel time per vehicle [sec]
$\pm 10\%$	NMPC	161	2.5	6.8	216
	RMPC	159	2.4	6.9	213
	Change	-1.2%	-4.0%	+1.5%	-1.4%
$\pm 20\%$	NMPC	173	2.6	6.4	230
	RMPC	167	2.5	6.6	223
	Change	-3.5%	-3.8%	+3.1%	-3.0%
$\pm 30\%$	NMPC	189	2.9	5.8	251
	RMPC	173	2.6	6.3	232
	Change	-8.5%	-10.3%	+8.6%	-7.6%

## 5 Conclusion

Nowadays, the work with computer simulations constitutes an important part of traffic engineering practice. The paper is based on the simulator VISSIM and the mathematical software MATLAB as they are widely used and accepted by engineers and researchers as well. By collecting the most important properties related to VISSIM COM interface and VISSIM API programming, an integrated VISSIM-MATLAB traffic simulation environment was introduced. To represent the applicability a robust control strategy was presented. In the application example COM interface was used for control of random traffic flows in VISSIM and DLL API was programmed together with MATLAB Engine for optimal signal control. The evaluation and validation were carried out by the use of the proposed simulation environment.

The authors hope that the results of the paper may provide an efficient help to traffic practitioners and developers.

## References

- 1 **Aboudolas K, Papageorgiou M, Kosmatopoulos E**, *Store-and-forward based methods for the signal control problem in large-scale congested urban road networks*, Transportation Research Part C: Emerging Technologies **17** (2009), 163–174, DOI 10.1016/j.trc.2008.10.002.
- 2 **Aboudolas K, Papageorgiou M, Kouvelas A, Kosmatopoulos E**, *A rolling-horizon quadratic-programming approach to the signal control problem in large-scale congested urban road networks*, Transportation Research Part C: Emerging Technologies **18** (2010), 680–694, DOI 10.1016/j.trc.2008.10.002.
- 3 **Box D**, *Essential COM*, Addison-Wesley, London, 1988. ISBN 0-201-63446-5.
- 4 **de Oliveira L B, Camponogara E**, *Multi-agent model predictive control of signaling split in urban traffic networks*, Transportation Research Part C: Emerging Technologies **18** (2010), no. 1, 120–139, DOI 10.1016/j.trc.2009.04.022.
- 5 **García C E, Prett D M, Morari M**, *Model predictive control: Theory and practice - A survey*, Automatica **25** (1989), 335–348, DOI 10.1016/0005-1098(89)90002-2.
- 6 **Gazis D C, Potts R B**, *The oversaturated intersection*, Second International Symposium on Traffic Theory (London, UK, 1963), pp. 221–237.

- 7 **Ghaoui L E, Oustry F, Lebrete H**, *Robust Solutions to Uncertain Semidefinite Programs*, SIAM Journal on Optimization **9** (1998), no. 1, 33–52, DOI 10.1137/S1052623496305717.
- 8 **Kulcsár B, Bécsi T, Varga I**, *Estimation of dynamic origin destination matrix of traffic systems*, Periodica Polytechnica ser. Transp. Eng **33** (2005), no. 1–2, 3–14.
- 9 **Löfberg J**, *Minimax approaches to robust model predictive control*, PhD thesis, Linköping University, Linköping, Sweden, 2003.
- 10 **Löfberg J**, *YALMIP: A Toolbox for Modeling and Optimization in MATLAB*, CACSD Conference (Taipei, Taiwan, 2004), pp. 284–289.
- 11 *MATLAB 7 External Interfaces*, The MathWorks, Inc, USA, 2007.
- 12 *User Manual for the VISSIM COM interface 5.2*, PTV Planung Transport Verkehr AG, Germany, 2010.
- 13 *VISSIM 5.2 User Manual*, PTV Planung Transport Verkehr AG, Germany, 2010.
- 14 **Tettamanti T, Varga I**, *Traffic control designing using model predictive control in a high congestion traffic area*, Periodica Polytechnica ser. Transp. Eng. **37** (2009), no. 1–2, 3–8, DOI 10.3311/pp.tr.2009-1-2.01.
- 15 **Tettamanti T, Varga I**, *Distributed traffic control system based on model predictive control*, Periodica Polytechnica ser. Civil. Eng. **54** (2010), no. 1, 3–9, DOI 10.3311/pp.ci.2010-1.01.
- 16 **Tettamanti T, Varga I, Kulcsár B, Bokor J**, *Model predictive control in urban traffic network management*, 16th Mediterranean Conference on Control and Automation (Ajaccio, Corsica, France, 2008), pp. 1538–1543.
- 17 **Welch G, Bishop G**, *An Introduction to the Kalman Filter*, TR95-041, University of North Carolina at Chapel Hill, 1995.
- 18 **Wiedemann R**, *Simulation des Straßenverkehrsflusses*, Schriftenreihe des Instituts für Verkehrswesen der Universität Karlsruhe **8** (1974).