Available online at www.sciencedirect.com

ScienceDirect

IFAC PapersOnLine 51-9 (2018) 416–421

Optimal headway merging for balanced public transport service in urban networks

Balázs Varga * Tamás Tettamanti * Balázs Kulcsár **

* Department of Control for Transportation and Vehicle Systems, Budapest University of Technology and Economics, Stoczek J. u. 2, H-1111, Budapest, Hungary (e-mail: varga.balazs@mail.bme.hu, tettamanti@mail.bme.hu)
** Department of Electrical Engineering, Chalmers University of Technology, Hörsalsvägen 9-11, SE-412-96, Gothenburg, Sweden (e-mail: kulcsar@chalmers.se)

Abstract: This paper presents a velocity control/advise algorithm relying on vehicle-to-vehicle communication, to ensure the headway homogeneity of buses on a joint corridor, i.e. when multiple lines merge and travel on the same route. The proposed control method first schedules merging buses prior to entering a common line. Second, based on the position and velocity of the bus ahead of the controlled one, a shrinking horizon model predictive controller (MPC) calculates a proper velocity profile for the merging bus. The model is able to predict short time-space behavior of public transport buses enabling constrained, finite horizon, optimal control solution to reach the merging point with equidistant headways, taking all buses from different lines into account. The controller is tested in a high fidelity traffic simulator with realistic scenarios.

Keywords: Headway homogeneity, Bus bunching, Automated transport systems, Model predictive control, Merging, Velocity control

1. INTRODUCTION

On busy lanes, where public transport buses are frequent, bus bunching is a common phenomenon. Due to bunching the periodicity of arrivals fail and homogeneous service cannot be provided (Sorratini et al. (2008)). In uncontrolled bus systems, bus bunching is prevalent especially in the peak hours. Newell and Potts (1964) point out that public transport have a natural tendency towards bunching and it is further worsened if multiple lines operate on a common route. Several approaches were proposed to deal with the problem of bus bunching. Daganzo and Pilachowski (2011) developed algorithms to control the headway of consecutive buses. Bartholdi and Eisenstein (2012) formulated a self controlling algorithm without timetable. Ampountolas and Kring (2015) proposed cooperative control of buses to mitigate bunching. Estrada et al. (2016) formulated a velocity control method considering bus-to-bus communication and green time extension. Andres and Nair (2017) used predictive methods to predict headways of consecutive buses.

The aforementioned works efficiently reduce bunching but focus on a single bus line exclusively. When bus lines merge, buses enter the common line according to their own schedule and the headway with the other line is not synchronized. It is desirable to introduce buses to the common route with equal headways to avoid bunching. Instead of including slack times as in Bartholdi and Eisenstein (2012) (for example at the first stop after the merging point), a gradually changing speed control is proposed: buses adjust their headways on the previous link, before entering the common line. In densely populated urban areas where city space is scarce, including slack times might not be possible due to bus stop configurations. Furthermore, slacks are an unproductive allocation of time of time in the cycle time of buses and results in queuing at stops (Daganzo (2009)). In these cases an adaptive velocity control is more desirable. The calculated velocity from the controller can be used by the driver or applied as a strict reference speed with the emergence of autonomous vehicles (Daganzo and Pilachowski (2011)).

Similar scheduling problems emerge in several fields: in industrial logistics where two conveyor systems merge or in package arbitration in data communications (Elahi et al. (2015); Athanasopoulou et al. (2013)). Merging problems are extensively studied in the context of highway on-ramps: Awal et al. (2013) develops vehicle-to-vehicle communication algorithm for optimal highway on-ramp traffic merging. Scarinci et al. (2013) proposes cooperative ramp metering control where vehicles communicate with each other and adjust their velocity via cruise control to help smooth merging. These works aim at merging two traffic streams in a microscopic manner. In the proposed bus merging strategy the controlled vehicles are far from each other, their instantaneous dynamics don't affect each other.

This paper purports to formulating a control algorithm that synchronizes the headways of buses on separate lanes before entering the common line. Since the buses are on different links, their headway cannot be defined. To this end a control strategy is proposed based on

2405-8963 © 2018, IFAC (International Federation of Automatic Control) Hosting by Elsevier Ltd. All rights reserved.
Peer review under responsibility of International Federation of Automatic Control.
10.1016/j.ifacol.2018.07.068

vehicle to vehicle communication and virtual headways. The proposed control strategy first determines in what order the buses can enter the common line. Then, based on the position and velocity of the leading bus a merging velocity is calculated, using a shrinking horizon model predictive control strategy which ensures equal headways upon entering the common line (Diehl et al. (2009)). The velocity control regulates headways only a few hundred meters upstream the junction on each leg, other bus operations are out of scope of this work.

The paper is organized as follows. In the Merging zones section the theoretical background of merging K bus lines is outlined and a control logic ensuring equal time headways is formulated. Then, in Section 3 a linear bus following model is proposed. Section 4 describes the shrinking horizon model predictive controller. Next, a simulation scenario is created in a high fidelity traffic simulator, serving as basis for the analysis of the control algorithm. In Section 6, simulation results are analyzed with different bus merging patterns. Finally, Section 7 concludes the findings of this paper.

2. MERGING ZONES

Buses operate on their dedicated lines with a desired periodicity. When these lines merge, they have to adhere to a different periodicity. Resulting offsets in the time headways and various arrival patterns shall be adjusted in order to remedy irregular service. In the followings a method is derived to ensure equidistant headway on joint bus corridors by means of optimized speed reference for merging.

When finite K bus lines merge, it is desirable to keep the periodicity of headways on the common line. To this end an algorithm is formulated which adjusts the velocity of the buses prior to entering the common line to ensure headway homogeneity. Here, headway refers to the time between consecutive buses. Lets call the point where these lines join merging point MP. In addition, define a merging area which starts a few hundred meters before the merging point on each upstream leg of the intersection. Denote the starting points of the merging area with E_i for each bus line i. If bus line $A, B, \ldots, K$ have headways $h_A, h_B, \ldots, h_K$ on their own routes respectively the headway on the common lane M, assuming equal headways, becomes:

$$h_M = \frac{\sum_{i=A}^{K} h_i}{K^2}. \tag{1}$$

In other words, the average headway is divided by the number of lines merging. Figure 1 depicts the merging problem for two bus lines.

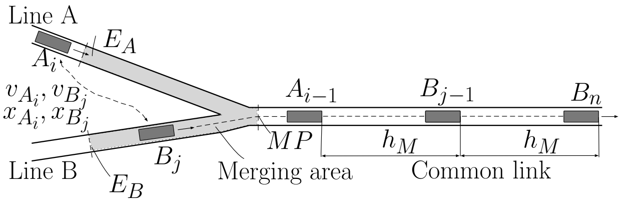

Fig. 1. Merging bus lines at a two-legged junction

Next, based on the timetable of individual bus lines, define a merging pattern P. It determines the order in which the buses shall travel on the common line. The simplest way to obtain such pattern is by taking the uncontrolled, periodic arrivals of each line at the merging point or at the first common stop. This inherently results in an arrival pattern but it can also be adjusted by the transport service provider. An example of three merging lines is presented in Figure 2. Timetable periodicity on individual lines result in a circular pattern, in Figure 2 two periods of P are shown. This pattern does not define the timetable exactly, only the order in which they shall operate on the common line. The pointer p shows the desired line to enter the common line. The control action is able to adjust the velocity of the buses in the merging area such that they reach the merging point MP according tho the desired pattern. The common timetable is the combination of P (order of buses) and h_M (headway).

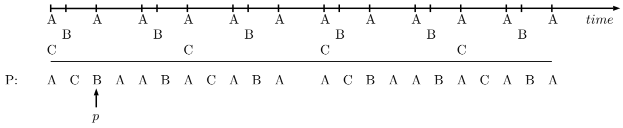

Fig. 2. Pattern of three merging buses with $h_A = 1$, $h_B = 2$ and $h_C = 3$.

Once buses enter the merging zone the velocity control can start. The flowchart of the velocity control is outlined in Figure 3.

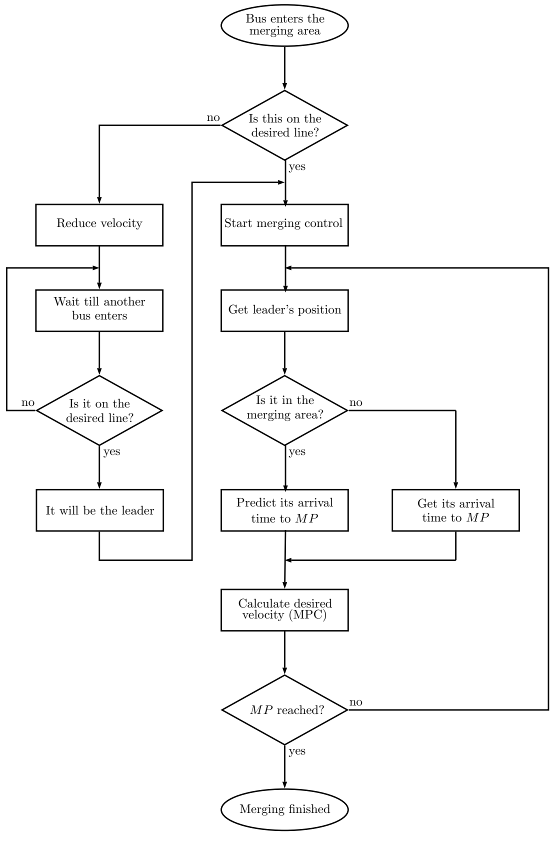
Bus enters the merging area
Is this on the desired line?
no
yes
Reduce velocity
Start merging control
Wait till another bus enters
Get leader's position
Is it on the desired line?
no
yes
Is it in the merging area?
no
yes
It will be the leader
Predict its arrival time to MP
Get its arrival time to MP
Calculate desired velocity (MPC)
MP reached?
no
yes
Merging finished

Fig. 3. Flowchart of the control algorithm for one bus (*this* denotes the controlled bus, *it* denotes the leading bus

If a bus i reaches the proximity (e.g. $500\ m$) of the start of the merging area E it is checked whether it can enter the common line. The line number of the bus $(A, B, \ldots, K)$ is compared to the desired one, defined by the p^{th} element of the pattern (P). If it matches, the velocity control starts and k is incremented by one. In case a bus arrives at E but its line number does not match $P(p)$, its velocity is reduced until another bus enters the merging area. It is then checked if this newly arrived bus can be the leader of the slowed down bus. This strategy makes it possible to reorganize buses in the merging area.

The velocity controller receives the position and moving average velocity of the leading vehicle x_{i-1} and $\bar{v}_{i-1}$, respectively. If the leading bus is still in the merging area $(x_{i-1} < MP)$ its arrival time to the merging point $t_{M,i-1}$ is extrapolated:

$$t_{M,i-1} = t_0 - \frac{MP - x_{i-1}}{\bar{v}_{i-1}}, \qquad (2)$$

where t_0 is the actual time instant. Since the buses might operate on separate lines their headway cannot be defined, extrapolation forms a virtual headway. The extrapolation does not consider obstacles in the merging area, such as traffic lights, intersections etc, only takes into account the moving average velocity of the bus. If the leading bus already left the merging area, extrapolation is not needed, $t_{M,i-1}$ can be directly forwarded to the controlled bus.

Figure 4 depicts the merging strategy in space-time diagram with three buses as example. The second bus B_j arrives at the merging area at $t_{E,j}$ after the leader bus A_{i-1} which already left it. The departure time of A_{i-1} from MP, $t_{M,i-1}$ is sent to B_j and the time interval $\Delta t_{i-1,j}$ and the desired velocity $v_{des,j}$ is calculated. After it leaves the merging area its desired velocity is set back to normal or determined by another control law. In Figure 4, when bus A_i arrives the bus ahead of it B_j is still in the merging area so $t_{M,j}$ is extrapolated.

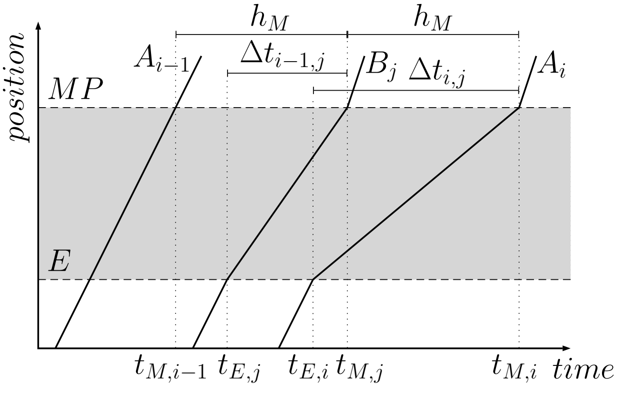

Fig. 4. Space-time diagram of the merging area

In the sequel the control oriented bus following model and the model predictive velocity controller is outlined, which ensures headway homogeneity on the common line.

3. BUS FOLLOWING MODEL FOR MERGING

The discrete-time model for the bus dynamics is based on the Optimal Velocity Model (OVM) (Bando et al. (1995)). Position $x(k)$, velocity $v(k)$ and acceleration $a(k)$ of a vehicle can be given as follows:

$$x(k+1) = x(k) + v(k)\Delta t, \qquad (3)$$

$$v(k+1) = v(k) + a(k)\Delta t, \qquad (4)$$

$$a(k) = \frac{1}{\beta}(v_{des}(k) - v(k)), \qquad (5)$$

where position $x(k+1)$ and velocity $v(k+1)$ denote the states over the time period of $[k\Delta t, (k+1)\Delta t]$ with discrete time step index k and sampling time Δt. $v_{des}(k)$ is the desired velocity at time step k and β is the relaxation term, a constant model parameter. The model is augmented with an additional equation, a position error denoting the distance from the merging point.

$$z(k+1) = x_{MP}(k) - x(k), \qquad (6)$$

the difference between the actual position of the bus $x(k)$ and a time dependent reference position $x_{MP}(k)$ defined by the merging point MP. Time dependency of $x_{MP}(k)$ will be detailed in Section 4.

The above equations can be written into state space form with $v_{des}(k)$ being the control input and $X(k) = [v(k), x(k), z(k)]^T$ the system states at time step k. The state space representation of the system is therefore:

$$\begin{bmatrix} v(k+1) \\ x(k+1) \\ z(k+1) \end{bmatrix} = \begin{bmatrix} 1 - \dfrac{\Delta t}{\beta} & 0 & 0 \\ \Delta t & 1 & 0 \\ 0 & -1 & 0 \end{bmatrix} \begin{bmatrix} v(k) \\ x(k) \\ z(k) \end{bmatrix} + \begin{bmatrix} \dfrac{\Delta t}{\beta} \\ 0 \\ 0 \end{bmatrix} v_{des}(k) + \begin{bmatrix} 0 \\ 0 \\ 1 \end{bmatrix} x_{MP}(k). \qquad (7)$$

4. SHRINKING HORIZON VELOCITY CONTROL DESIGN

The control oriented model outlined in Section 3 is used as basis of a shrinking horizon MPC design. The goal of the controller is calculating an optimal velocity profile between the actual position of the vehicle and the merging point, while the vehicle is inside the merging area. To this end, the position and velocity of the bus ahead is used.

Based on the leading bus' actual or extrapolated leaving time of the merging area $t_{M,i-1}$ and the time headway on the common route h_M, the merging time interval can be calculated:

$$\Delta t_{i-1,i}(k) = (t_{M,i-1} - t_0) + h_M. \qquad (8)$$

In other words $\Delta t_{i-1,i}(k)$ is the time frame in which the controlled bus shall reach MP.

Based on the discrete time step of the model τ and the time interval $\Delta t_{i-1,i}(k)$ the horizon length of the controller can be formulated:

$$\hat{H}_{i-1,i}(k) = \frac{\Delta t_{i-1,i}(k)}{\tau}. \qquad (9)$$

The horizon length of the controller $\hat{H}(k)$ is the desired arrival time to the end of the merging area MP.

In each time step the prediction horizon is recalculated. By the end of the horizon, the bus shall reach the merging point. To avoid small or even negative horizon lengths (due to lateness or being close to the merging point) a lower boundary for the horizon length is defined: $H_{min} = 5$. Too long horizon length shall also be avoided as it puts too much computational effort on the on-board control unit. To this end, an upper boundary $H_{max} = 100$ is also defined.

$$H(k) = max(H_{min}, min(\hat{H}(k), H_{max})). \tag{10}$$

The reference $x_{MP}(k)$ is constructed based on the unsaturated prediction horizon $\hat{H}(k)$:

$$x_{MP}(k) = x_0(k) + \frac{MP - x_0(k)}{\hat{H}(k)} j, \tag{11}$$

where $j \in 1, \dots, \hat{H}(k)$ is the prediction step. $x_{MP}(k)$ is the ideal trajectory between the current position of bus i and the desired arrival time at MP. In the calculation of $x_{MP}(k)$ the unsaturated prediction horizon $\hat{H}(k)$ is used. If the controller is unable to look ahead till MP, i.e. the horizon is too long ($H(k) > H_{max}$), the reference remains unchanged.

Next, the state space model in Equation (7) is extended for H horizon:

$$\overbrace{\begin{bmatrix} X(k+1|k) \\ X(k+2|k) \\ \vdots \\ X(k+H|k) \end{bmatrix}}^{\hat{\underline{\mathbf{x}}}} = \overbrace{\begin{bmatrix} A \\ A^2 \\ \vdots \\ A^H \end{bmatrix}}^{\underline{\underline{\mathcal{A}}}} \overbrace{X(k)}^{\underline{\mathbf{x}}}$$
$$+ \overbrace{\begin{bmatrix} B_u & 0 & \cdots & 0 \\ AB_u & B_u & & 0 \\ \vdots & \vdots & \ddots & \vdots \\ A^{H-1}B_u & A^{H-2}B_u & \cdots & B_u \end{bmatrix}}^{\underline{\underline{\mathcal{B}}}} \overbrace{\begin{bmatrix} u(k) \\ u(k+1|k) \\ \vdots \\ u(k+H-1|k) \end{bmatrix}}^{\underline{\mathbf{u}}}$$
$$+ \overbrace{\begin{bmatrix} E & 0 & \cdots & 0 \\ AE & E & & 0 \\ \vdots & \vdots & \ddots & \vdots \\ A^{H-1}E & A^{H-2}E & \cdots & E \end{bmatrix}}^{\underline{\underline{\mathcal{E}}}} \overbrace{\begin{bmatrix} x_{MP}(k) \\ x_{MP}(k+1|k) \\ \vdots \\ x_{MP}(k+H-1|k) \end{bmatrix}}^{\underline{\sigma}}. \tag{12}$$

Notations in equation (12) are summarized below:

- $X(k)$ is the vector of state variables: $X(k) = [v(k), x(k), z(k)]^T$.
- A denotes the state matrix.
- B_u is the control input matrix containing coefficients for the desired velocity.
- $u(k)$ is the controlled variable. The only control input to the system is the desired velocity of the bus $u(k) = v_{des}(k)$.
- E is the row selector matrix of the reference signal.
- x_{MP} is the reference signal.

The quadratic cost-function can be formulated with the help of the extended states:

$$J(k, H) = \frac{1}{2}\left[\hat{\underline{\mathbf{x}}}^T Q \hat{\underline{\mathbf{x}}} + \underline{\mathbf{u}}^T R \underline{\mathbf{u}}\right]. \tag{13}$$

$\hat{\underline{\mathbf{x}}}$ and $\underline{\mathbf{u}}$ denote stacked vectors of the predicted states (velocity, absolute and relative positions) and the control input (desired velocity) at each time step. Q and R are diagonal, positive semi-definite weighting matrices. A quadratic formula means that it penalizes both positive and negative deviations from the reference (i.e. not only late but early arrival too). R penalizes the control action, if R is high the system responds slowly. With some reformulation, the objective function to be minimized becomes:

$$\begin{aligned} J(k, H) = \frac{1}{2}\underline{\mathbf{u}}^T \overbrace{\left(\underline{\underline{\mathcal{B}}}^T Q\, \underline{\underline{\mathcal{B}}} + R\right)}^{\Phi} \underline{\mathbf{u}} \\ + \overbrace{\left(\underline{\mathbf{x}}^T \underline{\underline{\mathcal{A}}}^T Q\, \underline{\underline{\mathcal{B}}} + \underline{\sigma}^T \underline{\underline{\mathcal{E}}}^T Q\, \underline{\underline{\mathcal{B}}}\right)}^{\Omega^T} \underline{\mathbf{u}}. \end{aligned} \tag{14}$$

Finally, the control objective is

$$\min_u \left[\frac{1}{2}\underline{\mathbf{u}}^T \Phi \underline{\mathbf{u}} + \Omega^T \underline{\mathbf{u}}\right], \tag{15}$$

subject to:

$$|z(k + H|k)| < \varepsilon, \tag{16}$$
$$v_{min} \leq v_{des} \leq v_{max}. \tag{17}$$

In other words the position error shall be smaller than a few meters at the last time step, denoted by a parameter ε. Furthermore, it is assumed that the control input is bounded: $v_{min} = 10\ km/h$, $v_{max} = 50\ km/h$. The above optimization problem is a constrained quadratic programming problem.

5. SIMULATION SCENARIO

The velocity control algorithm is tested in a high-fidelity traffic simulator, VISSIM. The simulator can be used to generate different traffic scenarios and evaluate the developed control algorithm. A busy intersection in Budapest's residential area serves as basis of the analysis (Figure 5). Two bus lines 7 and 114 merge here. Buses travel in mixed traffic and the only obstacles are a stop and a traffic light for bus number 7. There are two lanes on both links, so slowed down buses can be overtaken. In the control

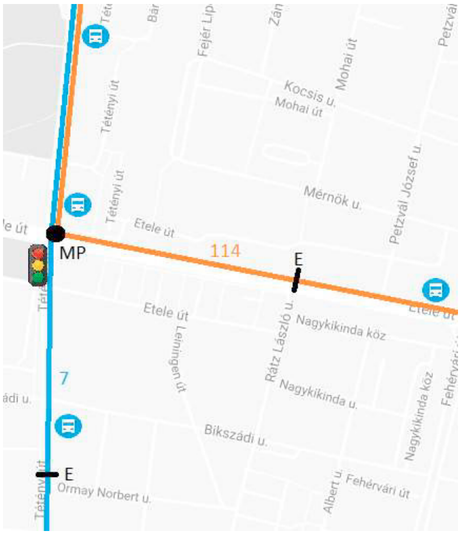

Fig. 5. Modeled junction, Budapest XI^{th} district, GPS coordinates of MP: 47.4638, 19.0337 (source: Google maps)

algorithm these obstacles are unaddressed, they act as perturbations to the system. Bus 7 and the common line have in-lane stops, 114 has a turnout. The length of the merging area in both legs of the junction is $480m$.

Time headway of bus 7 is $h_7 = 4\ min$ and the headway of 114 is $h_{114} = 5\ min$. The common headway becomes $h_M = 2.25\ min$. Furthermore, bus pattern on the common line is $P = [7,\ 114,\ 7,\ 114,\ 7,\ 144,\ 7,\ 7,\ 114]$.

6. SIMULATION RESULTS

In this section two simulation scenarios are provided. Each simulation is one hour long and presents the merging of the aforementioned bus lines 7 and 114. First, an uncontrolled scenario is given, without velocity control in the merging area. Figure 7 depicts bus trajectories in time-space diagram. Buses periodically arrive according to the schedule plus some normally distributed noise ranging from −30 to +30 seconds. Since there is no merging control buses merge irregularly, resulting in bunching or large headway gaps in the common line.

Next, merging control is introduced, see Figure 8. Buses on the separate lines arrive with their schedule on their own lines and start the merging process when reach the merging area (point E). Their velocity is adjusted so that they leave the junction with equal headways and according to pattern P. At 700 seconds bus 7 and 114 arrives at the merging area almost at the same time. According to P, the bus on line 114 shall go first so the bus on line 7 is slowed down. According to the pattern, this number 7 is followed by another bus 7.

Figure 6 shows the actual and desired velocity profiles of a selected bus on its route (line 114). The default desired velocity is 50 km/h and zero at stops outside the merging area, defined by the traffic simulator. Inside the merging area (marked by the dotted lines in the figure) the controller gradually reduces the desired velocity, delaying the bus. The desired velocity profile is tracked by the car following model built in the simulator, taking into account its surroundings (e.g. other vehicles, stops, etc.).

Table 1 compares the uncontrolled and the controlled simulation case based on statistical results. To be more accurate statistically, trajectories of multiple simulation runs are evaluated. Headways are compared at the exit point of the modeled network. The mean value is similar in both cases, close to the ideal headway of 270 seconds (2.25 min). Standard deviations, however, are significantly smaller in the controlled case. Finally, the Kullback-Liebler (KL) divergence is given between the ideal headway, and the simulation results, see Kullback and Leibler (1951). The ideal headway represents a uniform distribution with mean of 135 seconds and 0 variance. The KL distance is significantly smaller in the controlled case compared to the uncontrolled one, which means headways are more uniformly distributed.

Table 1. Statistics of the trajectories

	Mean (s)	Std (s)	KL distance
Uncontrolled case	130.06	73.57	0.172
Controlled case	135.83	32.16	0.024

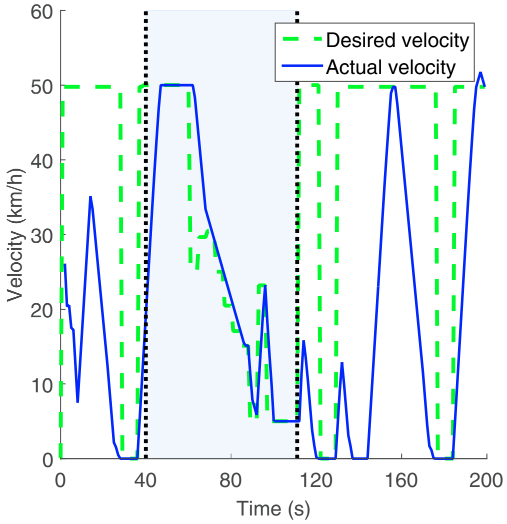

Fig. 6. Velocity profile of the 4^{th} bus 114. Time interval between 40 and 110 seconds is the merging intervention area.

7. CONCLUSION

In this paper a control algorithm was proposed to overcome bus bunching phenomenon happening on merging bus lines. After presenting the theoretical background of the merging of K bus lines, a control algorithm is formulated. The controlled bus selects the leader and receives its position and velocity. Based on this information it predicts its desired arrival time to the junction. The model predictive controller choses the desired velocity in the merging zone of the controlled buses in such a way that the buses enter the common line with equal time headways. The viability of the control algorithm is demonstrated with a high fidelity traffic simulator in a realistic scenario. Simulation results suggest that the velocity control can reduce bunching prior to bus lines merging. Using the KL divergence, it is shown that the control can guarantee more homogeneous headways. The algorithm is also able to reorganize buses in the merging area according to a predefined pattern.

ACKNOWLEDGEMENTS

This paper was supported by the ÚNKP-17-3 New National Excellence Program of the Ministry of Human Capacities of Hungary and by the János Bolyai Research Scholarship of the Hungarian Academy of Sciences.

B. Kulcsar acknowledges the support of Transport Area of Advance, a new initiative at Chalmers University of Technology, Gothenburg, Sweden.

REFERENCES

Ampountolas, K. and Kring, M. (2015). Mitigating bunching with bus-following models and bus-to-bus cooperation. In *IEEE 18th International Conference on Intelligent Transportation Systems (ITSC)*, 60–65. 15 - 18 September, 2015, Las Palmas, Spain.

Andres, M. and Nair, R. (2017). A predictive-control framework to address bus bunching. *Transportation Research Part B: Methodological*, 104, 123–148.

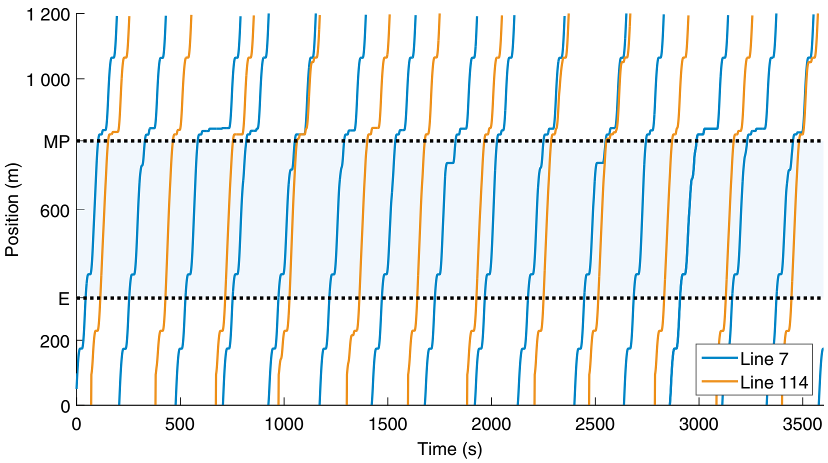

Fig. 7. Bus trajectories without merging control

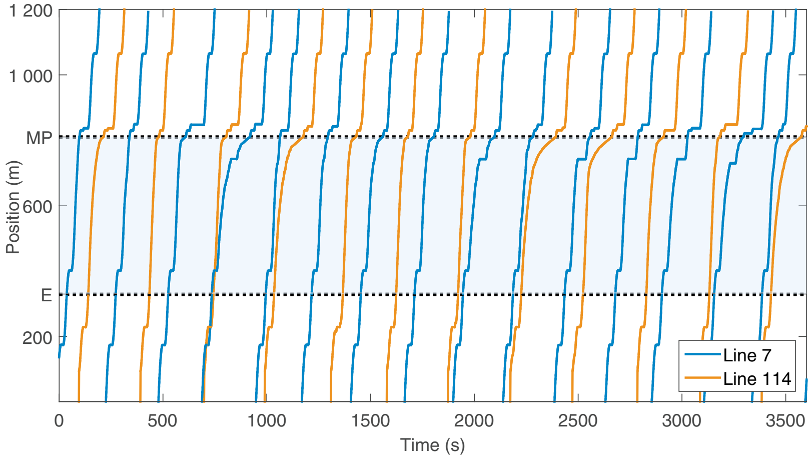

Fig. 8. Bus trajectories with merging control

Athanasopoulou, E., Bui, L.X., Ji, T., Srikant, R., and Stolyar, A. (2013). Back-pressure-based packet-by-packet adaptive routing in communication networks. *IEEE/ACM Transactions on Networking (TON)*, 21(1), 244–257.

Awal, T., Kulik, L., and Ramamohanrao, K. (2013). Optimal traffic merging strategy for communication-and sensor-enabled vehicles. In *16th International IEEE Conference on Intelligent Transportation Systems-(ITSC)*, 1468–1474.

Bando, M., Hasebe, K., Nakayama, A., Shibata, A., and Sugiyamai, Y. (1995). Dynamical model of traffic congestion and numerical simulation. *Physical review E*, 51(2), 1035–1043.

Bartholdi, J.J. and Eisenstein, D.D. (2012). A self-coordinating bus route to resist bus bunching. *Transportation Research Part B: Methodological*, 46(4), 481–491.

Daganzo, C.F. (2009). A headway-based approach to eliminate bus bunching: Systematic analysis and comparisons. *Transportation Research Part B: Methodological*, 43(10), 913–921.

Daganzo, C.F. and Pilachowski, J. (2011). Reducing bunching with bus-to-bus cooperation. *Transportation Research Part B: Methodological*, 45(1), 267–277.

Diehl, M., Ferreau, H.J., and Haverbeke, N. (2009). Efficient numerical methods for nonlinear MPC and moving horizon estimation. *Nonlinear model predictive control*, 384, 391–417.

Elahi, M.M.L., Rajpurohit, K., Rosenberger, J.M., Zaruba, G., and Priest, J. (2015). Optimizing real-time vehicle sequencing of a paint shop conveyor system. *Omega*, 55, 61–72.

Estrada, M., Mensión, J., Aymamí, J.M., and Torres, L. (2016). Bus control strategies in corridors with signalized intersections. *Transportation Research Part C: Emerging Technologies*, 71, 500–520.

Kullback, S. and Leibler, R.A. (1951). On information and sufficiency. *The annals of mathematical statistics*, 22(1), 79–86.

Newell, G.F. and Potts, R.B. (1964). Maintaining a bus schedule. In *2nd Conference of Australian Road Research Board (ARRB)*, volume 2. Melbourne, Australia.

Scarinci, R., Heydecker, B., and Hegyi, A. (2013). Analysis of traffic performance of a ramp metering strategy using cooperative vehicles. In *16th International IEEE Conference on Intelligent Transportation Systems-(ITSC)*, 324–329.

Sorratini, J.A., Liu, R., and Sinha, S. (2008). Assessing bus transport reliability using micro-simulation. *Transportation Planning and Technology*, 31(3), 303–324.