Hybrid Model Predictive Control of Public Transport Operations

Işık İlber Sırmatel

Nikolas Geroliminis

Swiss Federal Institute of Technology in Lausanne **May 2018**

STRC | 18th Swiss Transport Research Conference
Monte Verità / Ascona, May 16 – 18, 2018

Swiss Federal Institute of Technology in Lausanne

Hybrid Model Predictive Control of Public Transport Operations

Işık İlber Sırmatel
Urban Transport Systems Laboratory
Swiss Federal Institute of Technology in Lausanne
GC C2 385, Station 18, 1015 Lausanne
phone: +41-21-693 24 84
fax: +41-21-693 50 60
isik.sirmatel@epfl.ch

Nikolas Geroliminis
Urban Transport Systems Laboratory
Swiss Federal Institute of Technology in Lausanne
GC C2 389, Station 18, 1015 Lausanne
phone: +41-21-693 24 81
fax: +41-21-693 50 60
nikolas.geroliminis@epfl.ch

May 2018

Abstract

Bus transport systems cannot retain headways without feedback control due to their unstable nature, leading to irregularities such as bus bunching, and ultimately to increased travel times and decreased bus service quality. Development of feedback control methods for bus transport systems to avoid bus bunching and improve performance are thus of high importance. In this paper we propose a hybrid model predictive control (MPC) scheme that can regularize spacings and improve bus service quality. Performance of the predictive controller is compared with I- and PI-controllers via simulations. Results indicate the potential of the hybrid MPC in avoiding bus bunching, regularizing spacings and headways, and decreasing total service times.

Keywords

Public transport systems, bus speed control, bus bunching, hybrid systems, model predictive control.

1 Introduction

It is well known in the public transport systems literature that bus systems cannot maintain schedule without control (Newell and Potts, 1964). Buses that lag behind encounter more passengers waiting for them, leading to them lagging more, and buses that are slightly fast encounter less passengers. This positive feedback loop leads to the well-known phenomenon of bus bunching. Instabilities in the bus transport system (BTS) operation, resulting from spatiotemporal variability of both traffic congestion and stop-to-stop passenger demands, and manifesting themselves as headway irregularity and ultimately as bus bunching, lead to inefficient operations, increased service times, and degradation of service quality. Due to these reasons, research on designing feedback control systems for BTSs is of high importance.

Considerable research has been directed, especially in the last 4 decades, to developing real-time bus control methods for avoiding bus bunching and ensuring efficient and reliable operation of BTSs (see Ibarra-Rojas *et al.* (2015) and Sánchez-Martínez *et al.* (2016) for detailed reviews, and Berrebi *et al.* (2017) for an extensive review focusing on holding methods).

Most of the literature on bus operations via real-time control focus on station control methods, which involve taking decisions at a subset of stops of the bus loop. Some methods of this class focus on regularizing headways via holding, with the assumption that this would lead to efficient operation and decreased travel times (Abkowitz and Lepofsky, 1990, Daganzo, 2009, Xuan *et al.,* 2011, Andres and Nair, 2017). In situations where there is high variability in the demands, passenger waiting times need to taken into account in the holding problem formulation alongside headways (Ibarra-Rojas *et al.,* 2015). A recent study by Berrebi *et al.* (2015) considers stochasticity in bus arrival times and derives an optimal holding policy for minimizing headway irregularity by assuming that the distribution of bus arrivals is known and not influenced by decisions further upstream. Holding can also be used to improve timing of passenger transfers (Hall *et al.,* 2001, Delgado *et al.,* 2013). Another subclass of station control methods is the stop-skipping strategies, where the control decisions are realized by forcing buses to skip some stops, to increase speed and thus efficiency (Fu *et al.,* 2003). Although station control strategies can be effective in regularizing headways in moderate demand situations, for high demands they have adverse affects on BTS performance as they actuate via holding the bus at a stop or making the bus skip a stop. Under some circumstances they can make buses significantly slower, which will influence the quality of in-vehicle service, but also increase the operating cost and required fleet size. Another disadvantage of station control methods is that decisions can be taken only at stops, resulting in a significant time lag between observations and control actions. This can play a vital role if the system experiences various uncertainties both in time and space, which is the case en route (congestion heterogeneity) and at stops (passenger demand heterogeneity).

Another direction for control of public transport operations is the bus speed control, which involves manipulating the speed of each bus in real-time during its movement via feedback control mechanisms to avoid bunching and increase BTS efficiency. On this direction, a control strategy combining bus speed control and signal priority is developed in Chandrasekar *et al.* (2002), where control actions are taken to ensure that the buses operate with spacings equal to a desired spacing, which is shown to be able to regularize headways. A bus speed control method is proposed by Daganzo and Pilachowski (2011), where the speed command for each bus is computed according to its forward and backward spacings, which can enforce speed bounds and prevent bus bunching. A more recent study by Ampountolas and Kring (2015) develops a combined state estimation and linear quadratic regulator scheme to achieve coordination between the buses, leading to headway regularity and improved service.

The following points are crucial for real-time control of bus operations: (a) Constraints on speeds and passenger capacities of buses, (b) hybrid/mixed logical dynamical (MLD) phenomena (e.g., a bus can either cruise or stop while passengers can transfer only if a bus is stopping at the stop), (c) possibility of access to demand and traffic information without perfect knowledge. Considering these points, model predictive control (MPC) emerges as a control design paradigm highly applicable to control of bus operations. Based on real-time repeated optimization, MPC is an advanced control technique suited to optimal control of constrained multivariable nonlinear systems (note that MPC is also known as receding horizon control and rolling horizon planning/control). Main features of MPC are discussed in Garcia *et al.* (1989), whereas Mayne *et al.* (2000) provides an overview of theoretical aspects. A framework for hybrid MPC design is proposed in Bemporad and Morari (1999).

Motivated by the aforementioned points, in this paper we propose a hybrid MPC scheme considering a simplified MLD model with actuation via bus speeds, based on a mixed integer quadratic programming (MIQP) formulation to regularize bus spacings and obtain fast BTS operation. The method addresses the slowing-down problem of spacings-based bus speed control literature by integrating bus speed maximization in the problem formulation. Moreover, the proposed MIQP formulation yields convex quadratic programming (QP) subproblems (by construction due to the MLD modeling approach) when the integrality constraints are dropped. Such MIQPs, although still non-convex and NP-hard, can be solved much more efficiently compared to the more general integer/mixed integer NLPs with non-convex subproblems, since there exists powerful algorithms for solving the convex QP subproblems (Bonami *et al.,* 2012). Thus, the aforementioned feature of the proposed approach enables solutions of the hybrid MPC problems to global optimality while retaining real-time tractability.

2 Control Design

Achieving schedule reliability is an important goal in BTS control. This corresponds to minimizing the standard deviation of headways, as headways are directly related with waiting times. Nevertheless, headways are cumbersome to be integrated into a control framework. Regularizing bus spacings can be used as a proxy for achieving regular headways, as an ideal BTS condition where speeds and spacings of all buses are equal corresponds to perfect headway regularity. Still, under fast evolving conditions, this might not be the case. Although describing the evolution of headways is straightforward from a modeling point of view, the nonlinearity makes the integration in a control framework difficult. We consider this as a future direction of research.

In the following subsections we first describe in detail the main contribution of the paper: A novel hybrid MPC scheme that optimizes future BTS trajectories over bus speeds by solving an MIQP in real-time considering a weighted sum of two terms related to spacing regularization and fast BTS operation as objective function. We also present in a later subsection two bus speed controllers for comparison purposes: (i) An integral controller that reacts to spacing errors that is conceptually similar to the spacings-based controller proposed by Daganzo and Pilachowski (2011), (ii) a proportional-integral controller that reacts to both the spacing error and its rate of change, specifying a straightforward improvement over the integral controller.

2.1 Hybrid Model Predictive Control

We propose here a hybrid MPC with bus speed actuation considering BTS dynamics involving interactions between bus motion and passenger transfers between buses and stops, considering a BTS with K_b buses and K_s stops. The prediction model of the hybrid MPC includes all K_b buses operating on the loop but only $\tilde{K}_s(t)$ active stops (a subset of all stops, totaling K_s) that are upcoming stops for the buses, with $1 \leq \tilde{K}_s(t) \leq K_b$. In general, the number of active stops $\tilde{K}_s(t)$ is time-varying, as it is possible that there are two buses cruising to the same stop at the same time, in which case $\tilde{K}_s(t)$ would be smaller than K_b. Nevertheless, in normal conditions with well maintained spacings (such as those reported in the simulation results), and for a realistic BTS where the number of stops is larger than the number of buses, $\tilde{K}_s(t)$ is always equal to K_b. In an extreme (catastrophic) case in which all buses are cruising to the same stop then $\tilde{K}_s(t)$ would be 1. Such a model expresses a simplified model of the full BTS dynamics: The finite time horizon is expressed through the prediction horizon N, whereas the finite horizon in space is realized by considering only the first upcoming stop for each bus and omitting the rest.

2.1.1 Dynamics of Continuous States

(a) Predicted bus position dynamics can be written as:

$$x_i(k+1) = x_i(k) + T \cdot z_i(k) \tag{1}$$

where k is the prediction time step counter, whereas $x_i(k) \in \mathbb{R}$ and $z_i(k) \in \mathbb{R}$ are the predicted position and active speed of bus i, respectively.

(b) Dynamics of predicted bus accumulation states $n_i(k)$ (total accumulation on bus i) and $n_{i,a}(k)$ (the part of $n_i(k)$ with destination stop a) can be written as follows (analogous to equation (**??**)):

$$\begin{aligned} n_i(k+1) &= n_i(k) + T \cdot \left(q_i^{\text{in}}(k) - q_i^{\text{out}}(k)\right) \\ n_{i,a}(k+1) &= n_{i,a}(k) - T q_i^{\text{out}}(k), \end{aligned} \tag{2}$$

where $q_i^{\text{in}}(k) \in \mathbb{R}$ and $q_i^{\text{out}}(k) \in \mathbb{R}$ are the predicted boarding and alighting passenger flows, respectively, transferring between stop a and bus i.

(c) Dynamics of the predicted stop accumulation state $m_a(k)$ can be written as follows:

$$m_a(k+1) = m_a(k) + T \cdot \left(\beta_a(k, t_c) - \sum_{i=1}^{\mathcal{I}^a} q_i^{\text{in}}(k)\right), \tag{3}$$

where $\beta_a(k, t_c) \in \mathbb{R}$ is the passenger flow demand accumulating at stop a estimated at t_c, whereas $\mathcal{I}^a$ is the set of buses sharing stop a as their first upcoming stop.

2.1.2 Constraints Defining Binary Events

(a) The event $e_i^x(k) \in \mathbb{B}$ expresses whether bus i has reached stop a or not, and is defined as:

$$e_i^x(k) = \begin{cases} 1 & \text{if } 0 \leq x_i(k) \\ 0 & \text{otherwise.} \end{cases} \tag{4}$$

(b) The event $e_i^m(k) \in \mathbb{B}$ describes whether there are any passengers at stop a or not:

$$e_i^m(k) = \begin{cases} 1 & \text{if } m_a(k) \leq 0 \\ 0 & \text{otherwise.} \end{cases} \tag{5}$$

(c) The event $e_i^c(k) \in \mathbb{B}$ expresses whether bus i has available space for passengers or not:

$$e_i^c(k) = \begin{cases} 1 & \text{if } n_{i,\max} - n_i(k) \leq 0 \\ 0 & \text{otherwise.} \end{cases} \tag{6}$$

(d) The event $e_i^n(k) \in \mathbb{B}$ describes whether there are passengers on bus i that want to alight at stop a or not:

$$e_i^n(k) = \begin{cases} 1 & \text{if } n_{i,a}(k) \leq 0 \\ 0 & \text{otherwise.} \end{cases} \tag{7}$$

2.1.3 Dynamics of Binary States

(a) Predicted cruising state $\gamma_i(k) \in \mathbb{B}$ expresses whether bus i is cruising to stop a or not, and evolves according to following dynamics:

$$\gamma_i(k+1) = \gamma_i(k) \wedge \neg e_i^x(k). \tag{8}$$

(b) The predicted stopping state dynamics can be written as:

$$\delta_i(k+1) = \lambda_i(k) \vee \theta_i(k), \tag{9}$$

where $\delta_i(k) \in \mathbb{B}$ is the stopping state which expresses whether bus i is stopping at its upcoming stop a ($\delta_i(k) = 1$) or not ($\delta_i(k) = 0$), $\lambda_i(k) \in \mathbb{B}$ is the logical condition stating whether bus i begins stopping at stop a ($\lambda_i(k) = 1$) or not ($\lambda_i(k) = 0$), whereas $\theta_i(k) \in \mathbb{B}$ is the logical condition stating whether bus i continues stopping at stop a ($\theta_i(k) = 1$) or not ($\theta_i(k) = 0$). The predicted *begin stopping* condition $\lambda_i(k)$ is defined as:

$$\lambda_i(k) \triangleq \gamma_i(k) \wedge e_i^x(k) \tag{10}$$

which can be physically described as follows: Bus i begins stopping at stop a if it is cruising to stop a (i.e., if $\gamma_i(k) = 1$) and it reaches stop a (i.e., if $e_i^x(k) = 1$). Furthermore, the predicted *continue stopping* condition $\theta_i(k)$ is defined as:

$$\theta_i(k) \triangleq \delta_i(k) \wedge \neg((e_i^m(k) \vee e_i^c(k)) \wedge e_i^n(k)) \tag{11}$$

which can be physically described as follows: Bus i continues stopping at stop a if it is stopping (i.e., if $\delta_i(k) = 1$), and there are passengers wanting to alight (i.e., if $e_i^n(k) = 0$), or there are passengers wanting to board (i.e., if $e_i^m(k) = 0$) and the bus has vacant places (i.e., if $e_i^c(k) = 0$). Thus, equation (9) states that bus i begins stopping at stop a if the conditions for it to begin stopping are satisfied (i.e., $\lambda_i(k) = 1$), and it continues stopping at stop a as long as the conditions for its stopping remain satisfied (i.e., $\theta_i(k) = 1$). Once these former conditions are not satisfied anymore, stopping state of bus i becomes inactive (i.e., $\delta_i(k) = 0$) and it is thus allowed to move (representing its beginning to cruise to the stop after stop a) but it does not transition to a further cruising state since the stops other than the upcoming stop are not included in the prediction model.

2.1.4 Constraints on Bus Speed Control Inputs

The active speed of bus $z_i(k)$ is defined as:

$$z_i(k) = \begin{cases} u_i(k) & \text{if } \delta_i(k) = 0 \\ 0 & \text{otherwise,} \end{cases} \tag{12}$$

expressing the condition that bus i is restricted to have zero speed when it is stopping (i.e., when $\delta_i(k) = 1$), where $u_i(k)$ is the bus speed control input for bus i, which is bounded by the minimum and maximum speeds:

$$v_{\min} \leq u_i(k) \leq v_{a,\max}(k, t_c), \tag{13}$$

where $v_{a,\max}(k, t_c)$ is the maximum bus speed that depends on the traffic conditions at the link ending at stop a, and is thus to be estimated at each control time step t_c.

2.1.5 Constraints on Passenger Flows

(a) The predicted boarding passenger flow $q_i^{\text{in}}(k) \in \mathbb{R}$ transferring from stop a to bus i is defined through the following constraints:

$$\begin{aligned} q_i^{\text{in}}(k) &\leq \alpha \cdot \delta_i(k) \\ T \cdot q_i^{\text{in}}(k) &\leq n_{i,\max} - n_i(k) \\ T \cdot q_i^{\text{in}}(k) &\leq m_a(k). \end{aligned} \tag{14}$$

(b) The predicted alighting passenger flow $q_i^{\text{out}}(k) \in \mathbb{R}$ transferring from bus i to stop a is defined through the following constraints:

$$\begin{aligned} q_i^{\text{out}}(k) &\leq \alpha \cdot \delta_i(k) \\ T \cdot q_i^{\text{out}}(k) &\leq n_i(k). \end{aligned} \tag{15}$$

2.1.6 Initial States

Initial bus position state is defined as:

$$x_i(1) = -d_{i,a}(t_c) \tag{16}$$

where $d_{i,a}(t_c) \in \mathbb{R}$ is the distance of bus i to stop a measured at time t_c, whereas initial cruising state is:

$$\gamma_i(1) = \tilde{\gamma}_{i,a}(t_c), \tag{17}$$

where $\tilde{\gamma}_{i,a}(t_c)$ is the cruising state of bus i measured at t_c, and initial stopping state is

$$\delta_i(1) = \tilde{\delta}_{i,a}(t_c), \tag{18}$$

where $\tilde{\delta}_{i,a}(t_c)$ is the stopping state of bus i measured at t_c, and initial stop accumulation state is

$$m_a(1) = \sum_{j=1}^{K_s} \tilde{m}_{a,j}(t_c), \tag{19}$$

where $\sum_{j=1}^{K_s} \tilde{m}_{a,j}(t_c)$ is the total passenger accumulation at stop a measured at t_c, whereas initial bus accumulation states are

$$n_i(1) = \sum_{j=1}^{K_s} \tilde{n}_{i,j}(t_c) \tag{20}$$
$$n_{i,a}(1) = \tilde{n}_{i,a}(t_c),$$

where $\sum_{j=1}^{K_s} \tilde{n}_{i,j}(t_c)$ and $\tilde{n}_{i,a}(t_c)$ are the total passenger accumulation and the part destined to a inside bus i at measured t_c, respectively.

2.1.7 Hybrid Model Predictive Control Problem

We formulate the problem of finding the bus speed control input values that minimize a weighted sum of two terms related to spacing regularization and fast BTS operation as the following hybrid MPC problem:

$$\begin{aligned}
\underset{u_i(k)}{\text{minimize}} \quad & \sum_{i=1}^{K_b} \sum_{k=1}^{N} \left(e_i^2(k+1) + \sigma \cdot y_i^2(k) \right) \qquad (21) \\
\text{subject to} \quad & \text{for } i = 1, \ldots, K_b : \\
& \quad \text{initial states (17), (18), (19), (20)} \\
& \quad \text{for } k = 1, \ldots, N : \\
& \quad\quad 0 \le q_i^{\text{in}}(k) \\
& \quad\quad 0 \le q_i^{\text{out}}(k) \\
& \quad\quad \text{dynamics (1), (8), (9), (3), (2)} \\
& \quad\quad \text{constraints (12), (13), (14), (15)} \\
& \quad\quad \text{events (4), (5), (6), (7)}
\end{aligned}$$

where N is the prediction horizon, σ is the weight on fast operation term, whereas $e_i(k)$ and $y_i(k)$ are the predicted spacing and speed errors for bus i, respectively, and defined as follows:

$$e_i(k) = x_{i,f}(k) - 2x_i(k) + x_{i,r}(k) \tag{22}$$
$$y_i(k) = v_{a,\max}(k, t_c) - z_i(k),$$

where $x_{i,f}(k)$ and $x_{i,r}(k)$ are the predicted positions of the buses in front of and behind bus i, respectively. The problem (21) is an MIQP, which, although being a non-convex problem, can be solved efficiently (as it yields convex QPs when the integrality constraints are dropped) via software packages developed for mixed-integer programs.

2.2 Integral and Proportional-Integral Controllers

We first define front and rear spacings as follows:

$$s_{i,f}(t) = x_{i,f}(t) - x_i(t) \tag{23}$$
$$s_{i,r}(t) = x_i(t) - x_{i,r}(t) \tag{24}$$

where $x_{i,f}(t) \in \mathbb{R}$ and $x_{i,r}(t) \in \mathbb{R}$ are the positions of the buses in front of and behind bus i, whereas $s_{i,f}(t) \in \mathbb{R}$ and $s_{i,r}(t) \in \mathbb{R}$ are the front and rear spacings, respectively, with which we can define the spacing error $e_i(t) \in \mathbb{R}$ of bus i as follows:

$$e_i(t) = s_{i,f}(t) - s_{i,r}(t). \tag{25}$$

A discrete-time integral (I) controller with the goal of operating the BTS such that $e_i(t) = 0\ \forall i \in K_b$ can be formulated as follows:

$$u_i(t) = u_i(t-1) + K_{\mathrm{I}} \cdot e_i(t), \tag{26}$$

where $u_i(t) \in \mathbb{R}$ is the control input (bus speed command) for bus i and $K_{\mathrm{I}} \in \mathbb{R}$ is the integral gain. The I-controller simply updates the control input as a function of its spacing error $e_i(t)$, and how strongly it reacts to errors can be tuned by changing the K_{I} parameter. Note that the I-controller is conceptually similar to the bus speed controller proposed in Daganzo and Pilachowski (2011), in the sense that both controllers update bus speeds proportional to the spacing errors.

A straightforward improvement over the I-controller is the discrete-time proportional-integral (PI) controller that can be formulated as follows:

$$u_i(t) = u_i(t-1) + K_{\mathrm{P}} \cdot (e_i(t) - e_i(t-1)) + K_{\mathrm{I}} \cdot e_i(t), \tag{27}$$

where $K_{\mathrm{P}} \in \mathbb{R}$ is the proportional gain. The PI-controller updates the control input as a function of the spacing error $e_i(t)$ and its rate of change $e_i(t) - e_i(t-1)$, and how strongly it reacts to the two terms can be tuned by changing the K_{P} and K_{I} parameters.

While the I- and PI-controllers are easy to implement, they do not consider information about the number of passengers waiting at the stops or traveling on the buses, which might disturb the regularity of the spacings. Furthermore, there is no possibility of taking bus speed constraints explicitly into account with such controllers. Although computationally more cumbersome, the hybrid MPC specifies a solution for the aforementioned issues of the I- and PI-controllers.

3 Results

3.1 Results of a Congested Scenario

A bus loop with $K_b = 8$ buses and $K_s = 32$ stops is considered, where each link is 1 km long. The simulations are conducted using the BTS simulation model recently proposed in Sirmatel and Geroliminis (2017). Simulation sampling time is $T = 10$ s (which is found to yield a close match with smaller sampling times) and control sampling time is $T_c = 120$ s (i.e., the states are updated every 10 s whereas the bus speed control decisions are updated every 120 s), whereas simulation length is chosen as $t_{\text{final}} = 6480$ steps, which would correspond to 18 hours of real BTS operation. Passenger demands are constructed with morning and evening peaks, and demands to certain stops are chosen higher than the rest for capturing spatial variability and centers of attraction (i.e., some stops receive higher demands than others because they are more central). Bus capacity is $n_{i,\max} = n_{\max} = 80\ \forall i \in K_b$ passengers, whereas the maximum passenger flow parameter is $\alpha = 0.5$ passenger/s. Due to the formulation of the MLD simulation model there is no restriction on the number of births (i.e., the number of buses that can simultaneously serve a single stop). If a second bus arrives at a stop while the bus ahead of it is boarding passengers (specifying also a bus bunching event that would rarely happen if the BTS is operated with a bus control system), the passengers will begin boarding both buses. Absolute bounds on bus speeds are chosen as $v_{\min} = 4$ m/s and $v_{\max} = 20$ m/s, whereas the time-varying maximum link speeds $v_{j,\max}(t)$ are generated randomly (with a normal distribution) to simulate the effect of spatiotemporal variability of traffic conditions.

Controller gains of the I- and PI-controllers are selected as $K_{\text{P}} = 1.45$ and $K_{\text{I}} = 0.075$, while weighting factor of the hybrid MPC scheme is chosen as $\sigma = 10870$, which are found to give a good balance between regular headways and fast operation. The hybrid MPC scheme, with a prediction horizon of $N = 12$ (i.e., a horizon of 120 seconds as $T = 10$ s) is implemented using the YALMIP toolbox (Löfberg, 2004), in MATLAB 8.5.0 (R2015a) on a 64-bit Windows PC with 3.6-GHz Intel Core i7 processor and 16-GB RAM), with the MIQPs solved by calling Gurobi (Gurobi Optimization, 2016) from YALMIP. The prediction horizon value of $N = 12$ provides a good balance between performance and computational effort as it roughly covers the future time period in which buses arrive at their first upcoming stops, while it does not need to cover longer periods since the rest of the stops are not included in the prediction model as per the approximation of finite horizon in space. The hybrid MPC scheme is compared with the I- and PI-controllers. Simulation results with the congested scenario are given in Figs. 1 to 4, whereas a summary with numerical values of various performance criteria is given in Table 1. In Fig. 1 the bus positions (i.e., time-space diagrams) are shown for a period of simulation time (that

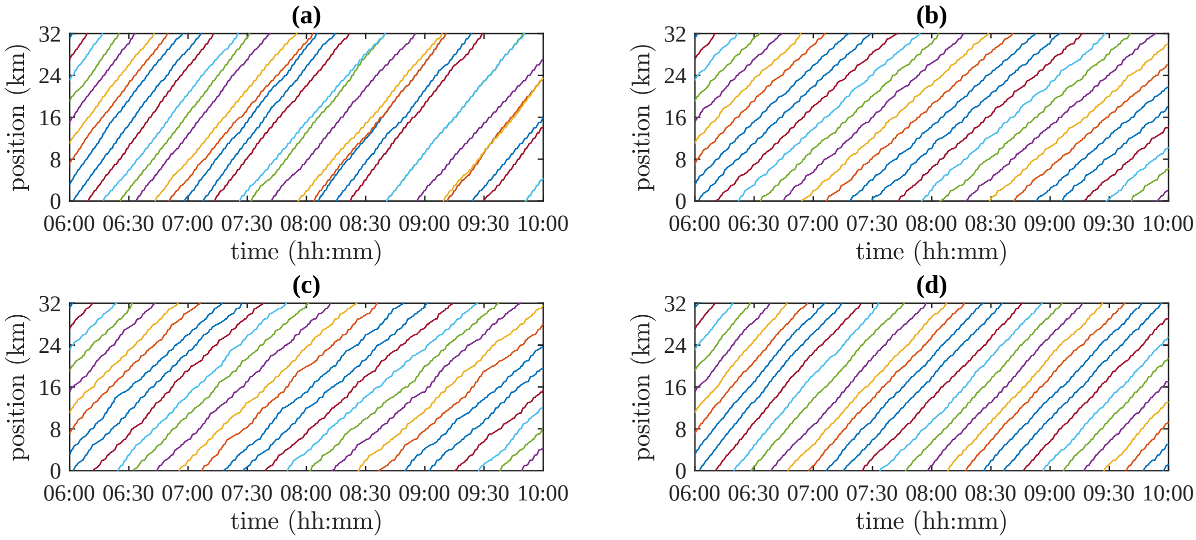

Figure 1: Bus positions for the first 4 hours of the congested scenario: (a) no control, (b) I-controller, (c) PI-controller, (d) hybrid MPC.

would correspond to the first 4 hours of real time) for the three proposed controllers, where each different color represents one bus. From Fig. 1 it can be seen that the no control case creates significant bus bunching phenomena, while all three controllers are able to regularize spacings. It is also clear that the hybrid MPC is superior to the others in the sense that it is also successful in operating the buses faster without any degradation to spacing regularity.

Figure 2 shows total demand $\sum_{h=1}^{K_s}\sum_{j=1}^{K_s}\beta_{h,j}(t)$ (the same for all controllers; zero values not shown), mean stop accumulation $\sum_{h=1}^{K_s}\sum_{j=1}^{K_s} m_{h,j}(t)/K_s$, mean commercial speed $\sum_{i=1}^{K_b} v_i(t)/K_b$, and mean bus accumulation $\sum_{i=1}^{K_b}\sum_{j=1}^{K_s} n_{i,j}(t)/K_b$, all as functions of time, comparing the three controllers for the whole simulation length (which would correspond to 18 hours of real time). Note that these curves are down-sampled in time with averaging in 5 minute intervals for clearer exposition. From the figure the effect of the morning and evening peaks in the demand on the BTS performance can be seen clearly: In periods of high demand all controllers suffer from decreased mean commercial speeds due to higher dwell times, leading to increased passenger accumulations. Hybrid MPC outperforms the other controllers, however, as it is able to maintain bus speeds that are around 16% larger than those of the I- and PI-controllers. This translates especially to comparatively lower levels of accumulations, especially at stops, ultimately yielding decreased service times.

Figure 3 shows the headway and spacing error distributions of the three controllers are shown, from which two observations can be made: (1) Regularizing spacings through feedback control leads to, as expected, headway regularity, as all controllers are able to achieve headway distributions that resemble normal distributions with reasonably low standard deviations, (2) hybrid MPC is able to decrease the mean of the headway distribution without any increase in its

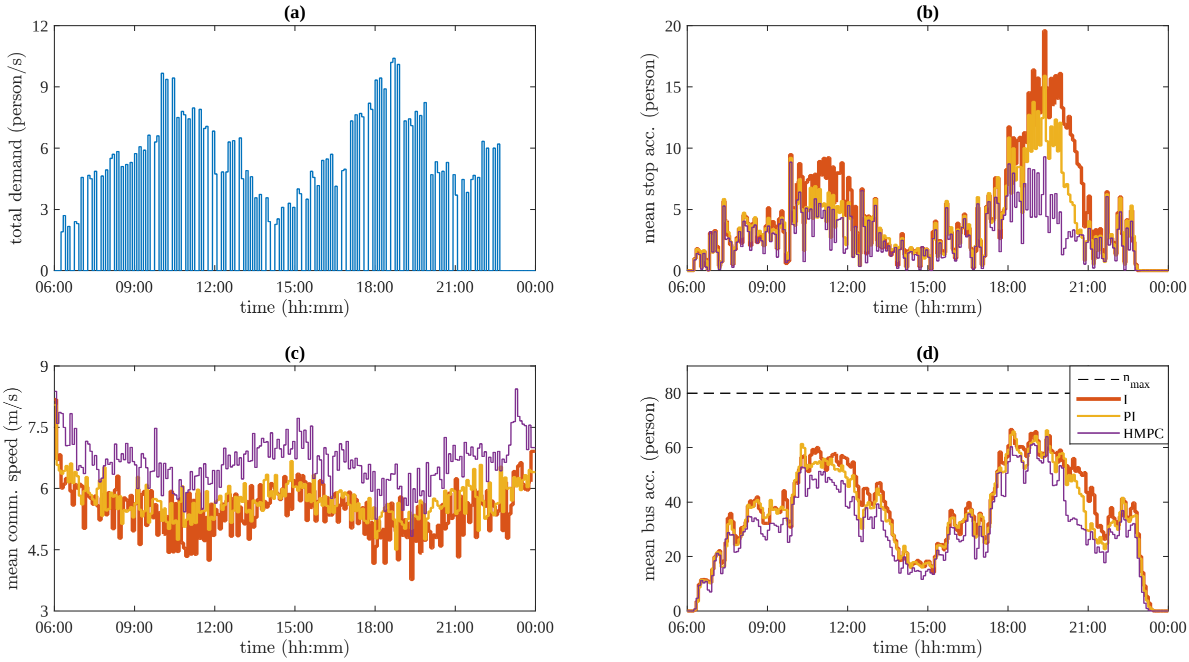

Figure 2: Total demand (a), mean stop accumulation (b), mean commercial speed (c), and mean bus accumulation (d), as functions of time, for the congested scenario.

standard deviation. From the latter it can be concluded that hybrid MPC can indeed increase BTS operation speed without harming headway regularity. This is mainly because the hybrid MPC is able to operate buses at higher speeds. Thus, equal spacing strategies might result in decreased performance in terms of bus speed and time spent at stops by passengers, as they overreact and slow down the buses significantly. It is also clear that minimizing spacing error is not equivalent to minimizing headway standard deviation.

Figure 4 shows the accumulations $n_i(t)$ for buses 1, 2, and 3 for the congested scenario, comparing the I- and PI-controllers and the hybrid MPC. From the figure it can be observed that, compared to the I- and PI-controllers, the hybrid MPC has a smaller amount of cases with full buses. Owing to the fact that it can serve the same demand faster (due to higher values of average commercial speeds without sacrificing headway regularity), the hybrid MPC is able to better keep the buses away from being completely full, resulting in a higher service quality.

Overall performances (mean time spent at stop (TSS) and in bus (TSB) per passenger, mean commercial speed (averaged over time and buses), mean and standard deviation of headways, standard deviation of spacing errors, and computation time for hybrid MPC) are summarized with the numerical values given in Table 1, where results of a no control (NC) case (i.e., in which the buses are operated at the maximum possible speed) are also included for comparison. The no control case suffers from bus bunching and thus experiences excessive TSS, although its TSB value is slightly lower in relation to the controlled cases owing to a larger mean commercial

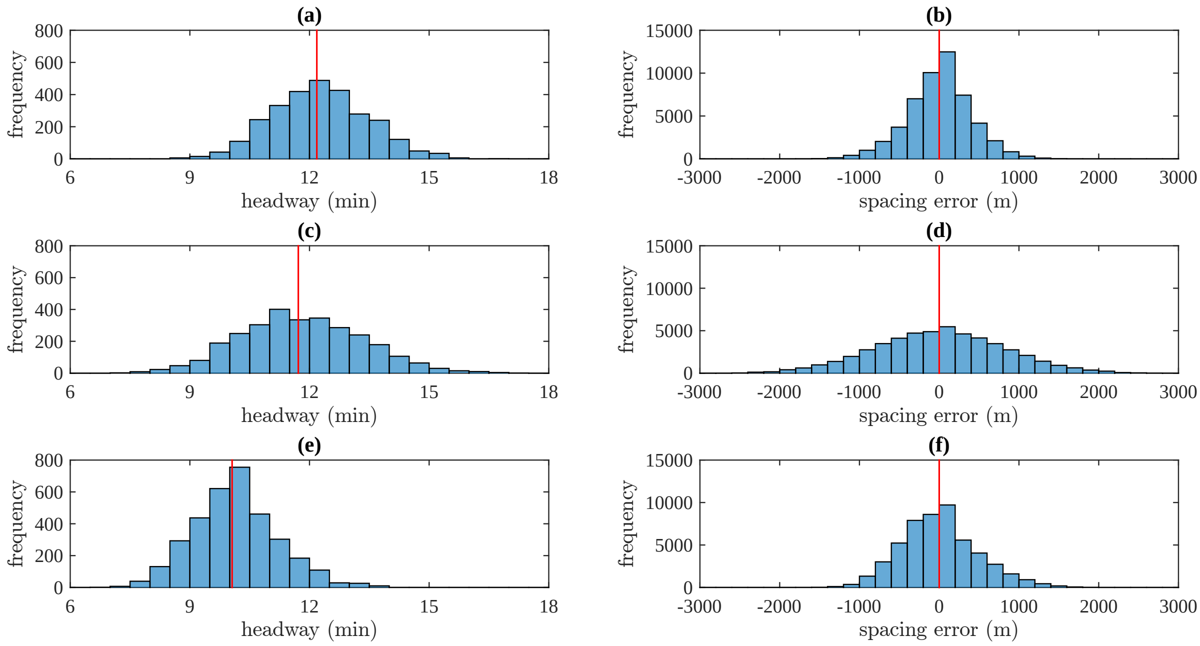

Figure 3: Headway ((a), (c), (e)) and spacing error ((b), (d), (f)) distributions of the congested scenario (vertical red lines show the mean values, which are 0 for spacing errors by definition): (a)-(b) I-controller, (c)-(d) PI-controller, (e)-(f) hybrid MPC.

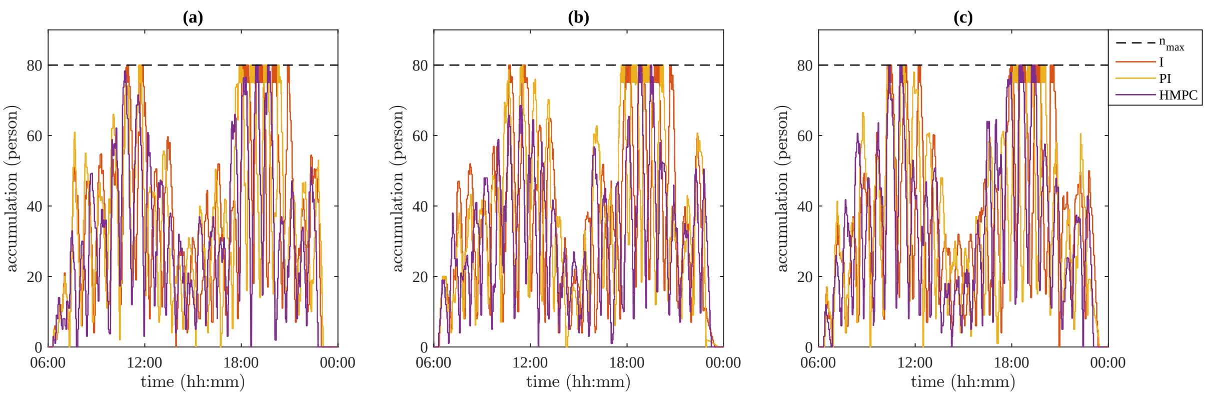

Figure 4: Accumulations of (a) bus 1, (b) bus 2, and (c) bus 3, for I- and PI-controllers and the hybrid MPC for the congested scenario.

speed. The I-controller yields a relatively slow operation leading to a higher value of TSS, but is able to achieve both spacing and headway regularity, with a standard deviation of spacing errors lower than that of other controllers. Nevertheless, as there is no one-to-one relationship between headways and spacings under dynamic conditions, achieving headway regularity is not guaranteed to result in low spacing errors. It is also clear that low spacing errors can result in slow-moving buses. Compared with the I-controller, the PI-controller is able to achieve a somewhat faster operation albeit with a trade-off in headway regularity, leading to slightly decreased values of TSS and TSB but increased standard deviation of headways. Hybrid MPC is seen to be superior to both controllers in both aspects of fast BTS operation (i.e., lower service times) and headway regularization (i.e., smaller standard deviation of headways): (a) It can decrease the total service time (TST, i.e., the sum of TSS and TSB) of passengers by 25% compared to the I- and 17% compared to the PI-controller, (b) it improves headway regularity by 11% compared to the I- and 31% compared to the PI-controller. Performance of the hybrid MPC is reflected in its smaller mean of headways value, owing to a faster operation, but also smaller standard deviation of headways, suggesting that it can improve these two conflicting objectives at the same time. In additional simulation studies, the effect of uncertainty on measurements of bus accumulations was examined by adding normally distributed noise to the $\tilde{n}_{i,a}(t_c)$ values with standard deviations up to 5 passengers: From the results it is observed that the performances in service times and standard deviation of headways show a deterioration up to only 1% compared to the noise-free case, showcasing the resilience of the hybrid MPC algorithm against errors in the measurement of $\tilde{n}_{i,a}(t_c)$. The average/maximum computation times of 0.3/0.7 s needed for solving one instance of the MIQP given in (21) indicate the computational tractability of the hybrid MPC, suggesting high potential for practical applications requiring real-time control decisions. One might argue that a different objective function such as minimum TST of all passengers could be employed. While this is straightforward to implement, the performance of such a controller is lower than the proposed hybrid MPC, which might look surprising. The main reason is that the prediction horizon (only one stop ahead) does not allow for a proper prediction of passenger travel times for the duration of the trip. A very long prediction horizon might be a solution, but this would lead to loss of real-time feasibility due to excessive computational burden. Examining this issue should be an interesting research priority.

Table 1: Control Performance Evaluation for a Congested Scenario

performance criterion	NC	I	PI	HMPC
mean time spent at stop (min)	21.8	9.4	7.6	5.6
mean time spent in bus (min)	13.5	18.7	17.8	15.5
mean commercial speed (m/s)	7.6	5.5	5.7	6.6
mean of headways (min)	–	12.2	11.7	10.1
standard deviation of headways (min)	–	1.19	1.53	1.06
standard deviation of spacing errors (m)	–	398	814	478
mean/max computation time (s)	–	–	–	0.3/0.7

3.2 Results of a Set of Congested Scenarios

The congested scenario results given in the previous section represent a single randomly generated scenario. As the passenger flow demands $\beta_{h,j}(t)$ and the maximum bus speeds $v_{i,\max}(t)$ are signals including random variables, a single random scenario might not be representative of the overall situation. Thus, to be able to draw a clearer picture regarding the performance differences between the controllers, a series of simulation experiments are conducted based on 100 randomly generated scenarios.

The results are summarized in Fig. 5, which shows the histograms of mean TST, standard deviation of headways, mean commercial speed, and standard deviation of spacing errors for the 100 scenarios, comparing the I- and PI- controllers with the hybrid MPC. The results emphasize the superiority of the hybrid MPC: On average, it can decrease TST around 20% against the I- and 16% against the PI-controller, at the same time showing an improvement in headway regularity around 2% against the I- and 30% against the PI-controller. The TST results are also interesting as they show that the hybrid MPC is more consistent in its performance regarding passenger delays: While the other controllers have TST values spread around relatively large intervals (roughly between 24 and 29 minutes for the I- and 23.5 and 27 minutes for the PI-controller), the hybrid MPC consistently performs in the 20.25-22 minute interval, showcasing the resilience of its performance in minimizing passenger delays against a diverse set of scenarios. The mean commercial speed results indicate that the hybrid MPC is able to retain better headway regularity even though it drives the buses faster (on average, 20% faster than the I- and 16% faster than the PI-controller) than the other controllers. A better performance in headway regularity is observed not to directly translate to a higher regularity in spacings, as seen from the spacing regularity figure. Furthermore, the I-controller is superior to hybrid MPC in terms of spacing regularity, which can be attributed to the fact that, based on its objective function, the hybrid MPC is forming a trade-off between fast operation and regular spacings, while the I-controller is operating only on spacing errors. In any case, passengers do not experience directly the effect of spacing regulation, what is important for them is time spent on board (i.e., TSB) and while waiting for a bus to arrive (i.e., TSS), and that buses arrive relatively regularly at the stop (i.e., a small standard deviation of headways). Both of these metrics are significantly better using the hybrid MPC. Overall, the hybrid MPC is successful in achieving a more desirable outcome in terms of practical performance, as what is important from both the passenger and the bus operator points of view is the minimization of TST and regularization of headways.

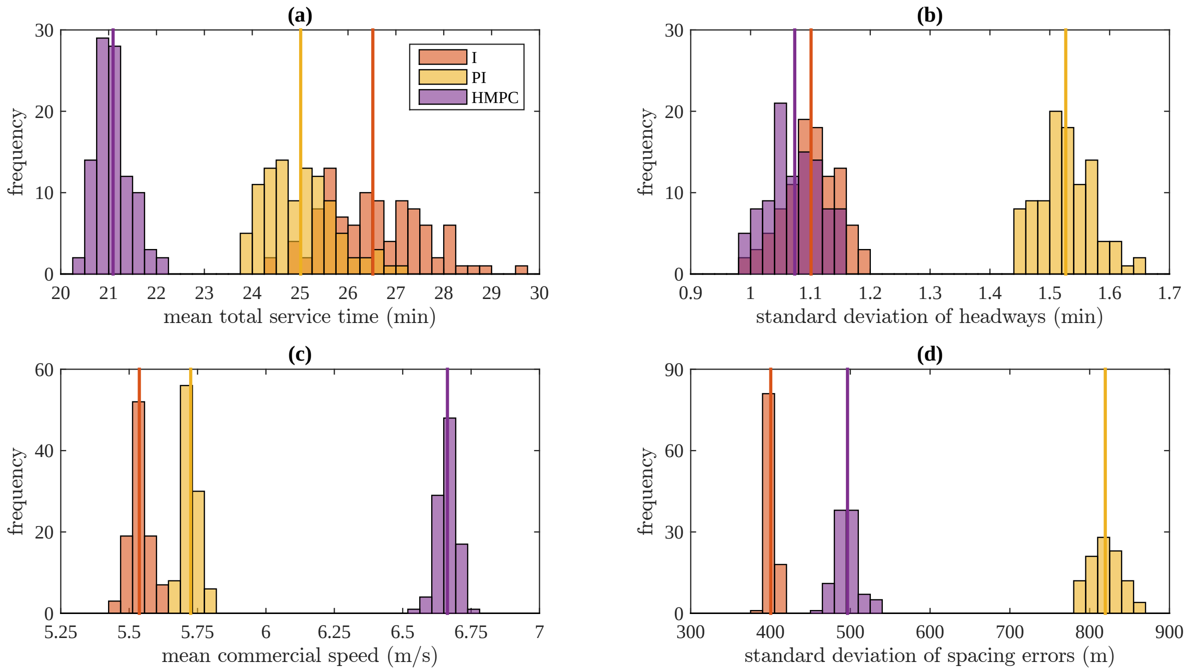

Figure 5: Histograms of the performance criteria from 100 randomly generated scenarios, comparing the I- and PI- controllers with the hybrid MPC: (a) mean total service time, (b) standard deviation of headways, (c) mean commercial speed, (d) standard deviation of spacing errors.

4 Conclusion

We developed a novel hybrid MPC scheme based on a simple MLD model for computational tractability. The hybrid MPC achieves consistently high BTS performance with a dual objective of spacing regularity and fast operation, by coordinating the buses operating on the loop via manipulating their speeds in real time. By construction, the proposed MPC formulation results in MIQPs having convex QP subproblems when the integrality constraints are dropped, enabling efficient solutions with computation times less than 0.7 seconds. Performance of the hybrid MPC is compared to classical I- and PI-controllers from literature via detailed simulations. Results indicate the potential of the hybrid MPC in decreasing service times without degradations in headway regularity. Performance and real-time tractability of the proposed hybrid MPC suggest high value for practical applications. Future work could include extensions of the model and controllers for multi-loop bus systems, establishing stability properties of the controllers, and comparisons of the proposed model with microscopic BTS simulation frameworks.

5 References

Abkowitz, M. D. and M. Lepofsky (1990) Implementing headway-based reliability control on transit routes, *Journal of Transportation Engineering*, **116** (1) 49–63.

Ampountolas, K. and M. Kring (2015) Mitigating bunching with bus-following models and bus-to-bus cooperation, paper presented at the *Intelligent Transportation Systems (ITSC), 2015 IEEE 18th International Conference on*, 60–65.

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

Bemporad, A. and M. Morari (1999) Control of systems integrating logic, dynamics, and constraints, *Automatica*, **35** (3) 407–427.

Berrebi, S. J., E. Hans, N. Chiabaut, J. A. Laval, L. Leclercq and K. E. Watkins (2017) Comparing bus holding methods with and without real-time predictions, *Transportation Research Part C: Emerging Technologies*.

Berrebi, S. J., K. E. Watkins and J. A. Laval (2015) A real-time bus dispatching policy to minimize passenger wait on a high frequency route, *Transportation Research Part B: Methodological*, **81**, 377–389.

Bonami, P., M. Kilinç and J. Linderoth (2012) Algorithms and software for convex mixed integer nonlinear programs, in *Mixed Integer Nonlinear Programming*, 1–39, Springer.

Chandrasekar, P., R. Long Cheu and H. C. Chin (2002) Simulation evaluation of route-based control of bus operations, *Journal of transportation engineering*, **128** (6) 519–527.

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

Delgado, F., N. Contreras and J. C. Munoz (2013) Holding for transfers, paper presented at the *Transportation Research Board 92nd Annual Meeting*, no. 13-2913.

Fu, L., Q. Liu and P. Calamai (2003) Real-time optimization model for dynamic scheduling of transit operations, *Transportation Research Record: Journal of the Transportation Research Board*, (1857) 48–55.

Garcia, C. E., D. M. Prett and M. Morari (1989) Model predictive control: theory and practice – a survey, *Automatica*, **25** (3) 335–348.

Gurobi Optimization, I. (2016) Gurobi optimizer reference manual, `http://www.gurobi.com`.

Hall, R., M. Dessouky and Q. Lu (2001) Optimal holding times at transfer stations, *Computers & industrial engineering*, **40** (4) 379–397.

Ibarra-Rojas, O., F. Delgado, R. Giesen and J. Muñoz (2015) Planning, operation, and control of bus transport systems: A literature review, *Transportation Research Part B: Methodological*, **77**, 38–75.

Löfberg, J. (2004) YALMIP: A toolbox for modeling and optimization in MATLAB, paper presented at the *IEEE International Symposium on Computer Aided Control Systems Design*, 284–289.

Mayne, D. Q., J. B. Rawlings, C. V. Rao and P. O. Scokaert (2000) Constrained model predictive control: Stability and optimality, *Automatica*, **36** (6) 789–814.

Newell, G. F. and R. B. Potts (1964) Maintaining a bus schedule, paper presented at the *Australian Road Research Board (ARRB) Conference, 2nd, 1964, Melbourne*, vol. 2, 388393.

Sánchez-Martínez, G., H. Koutsopoulos and N. Wilson (2016) Real-time holding control for high-frequency transit with dynamics, *Transportation Research Part B: Methodological*, **83**, 1–19.

Sirmatel, I. and N. Geroliminis (2017) Dynamical modeling and predictive control of bus transport systems: A hybrid systems approach, vol. 50, 7499 – 7504, ISSN 2405-8963. 20th IFAC World Congress.

Xuan, Y., J. Argote and C. F. Daganzo (2011) Dynamic bus holding strategies for schedule reliability: Optimal linear control and performance analysis, *Transportation Research Part B: Methodological*, **45** (10) 1831–1845.