https://www.sciencedirect.com/science/article/pii/S0191261518300468

https://doi.org/10.1016/j.trb.2018.06.009

Mixed Logical Dynamical Modeling and Hybrid Model Predictive Control of Public Transport Operations

Isik Ilber Sirmatel, Nikolas Geroliminis

Urban Transport Systems Laboratory (LUTS)
School of Architecture, Civil and Environmental Engineering
École Polytechnique Fédérale de Lausanne (EPFL)
1015 Lausanne, Switzerland

Abstract

Bus transport systems cannot retain scheduled headways without feedback control due to their unstable nature, leading to irregularities such as bus bunching, and ultimately to increased service times and decreased bus service quality. Traditional anti-bunching methods considering only regularization of spacings might unnecessarily slow down buses en route. In this work a detailed but computationally lightweight dynamical model of a single line bus transport system involving both continuous and binary states is developed. Furthermore a hybrid model predictive control (MPC) scheme is proposed, with a dual objective of regularizing spacings and improving speed of bus service operations. Performance of the predictive controller is compared with I- and PI-controllers via extensive simulations using the proposed model. Results indicate the potential of the hybrid MPC in avoiding bus bunching and decreasing passenger delays inside and outside the buses.

Keywords: Public transport systems, Bus speed control, Bus bunching, Mixed logical dynamical 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 modeling and control of BTSs is of high importance.

Modeling efforts to represent bus dynamics in congested routes have been investigated by various papers. The reader could refer to Hans et al. (2015a) for a review. The authors describe models of two main categories (deterministic and stochastic) and investigate how well they can reproduce service irregularities. Influences of overtaking and common lines on the performance of bus operations are examined in Schmöcker et al. (2016), whereas Wu et al. (2017) study the effects of dynamic passenger queue swapping considering bus bunching and capacity constraints. Recently, an interesting work (see Hans et al. (2015b)) develops a stochastic and event based bus operation model that provides predictions of bus trajectories based on the observation of current bus positions using particle filters. While the estimates are quite accurate, such a framework might be difficult to be integrated in a real-time control framework, due to the potentially prohibitive computational burden associated with such models.

An interesting feature of BTSs is the presence of hybrid dynamical phenomena, suggesting the necessity of modeling them as hybrid systems, where evolution of the dynamics depend on the interaction of continuous and discrete variables (Van Der Schaft and Schumacher, 2000). Buses can be cruising (i.e., can have nonzero speed) only if they are not stopping at a stop, whereas passengers can transfer between a stop and a bus only if the bus is stopping at that stop. As a consequence, two kinds of information are needed to describe BTS dynamics: Continuous variables (e.g., bus positions or passenger accumulations) and binary variables (e.g., the condition whether a bus is currently cruising to a stop). Evolution of continuous variables are subject to linear dynamics since bus motion and passenger accumulation dynamics can be formulated using linear models, whereas binary variables evolve according to a finite state machine (e.g., the condition whether a bus is currently cruising to a stop becomes false as the condition whether the bus is currently stopping at that stop becomes true when the bus reaches the stop). Such hybrid systems can be modeled as mixed logical dynamical (MLD) systems, which are hybrid systems with dynamics evolving according to linear difference equations subject to inequality constraints involving continuous and discrete variables. Proposed by Bemporad and Morari (1999), the MLD modeling framework is a systematic approach to the mathematical modeling of dynamical systems which involve the interaction of physical laws, logic rules, and constraints. Dynamics of a BTS feature interactions of continuous (e.g., bus positions and passenger accumulations) and binary states (e.g., is bus 1 stopping at stop 2?), involving both physical laws (e.g., buses moving or passengers flowing) and logic statements (e.g., passengers may transfer between bus 1 and stop 2 only if bus 1 is stopping at stop 2). Approaching the BTS modeling problem from a hybrid systems point of view is thus necessary for building detailed dynamical models of bus operations for analysis and control purposes, whereas the MLD modeling is preferable as it enables development of hybrid dynamical models with beneficial properties for reasons of computational efficiency.

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 line. 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 be 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 class of real-time bus control methods is the inter-station control, where decisions are taken while the bus is moving between stops. Traffic signal priority methods belong to this category, where the aim is to manage traffic flow efficiently via prioritizing certain circulations of an intersection with actuation over traffic lights (Liu et al., 2003; Van Oort et al., 2012; Chow et al., 2017). Although consistent with the standard framework where control inputs belong to the n-dimensional real space $\mathbb{R}^n$ that is prevalent in the control systems literature, bus speed control methods (another member of the inter-station control class) received relatively little attention compared to holding methods in the public transport literature. Signal priority is not studied in this work.

The idea in bus speed control is to manipulate 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 dynamical 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 method to design

MPC for hybrid systems (i.e., hybrid MPC) is proposed in Bemporad and Morari (1999) together with the MLD modeling framework.

Using MPC methods for bus operations received increased attention in the public transport literature during the recent years. In Eberlein et al. (2001), the holding problem is formulated as a deterministic MPC scheme to minimize passenger delays, using a model considering dwell time effects on vehicle delays and headways. The problem of minimizing passenger delays via stop-skipping is modeled using an integer programming formulation in Sun and Hickman (2005). Delgado et al. (2009) develop a model considering deterministic passenger demands at stops and travel times between stops, which is used to design an MPC with linear constraints for holding and boarding limits. A multi-objective hybrid MPC formulation is proposed in Cortés et al. (2010) using a dynamical model considering bus position, loads, and arrival times. The method considers holding and stop-skipping as control inputs to minimize waiting times and adverse effects of the control actions. Considering holding times and number of passengers prevented from boarding as control inputs, Delgado et al. (2012) integrate dynamics of bus positions and passenger accumulations in an MPC framework that involves minimizing passenger delays and control action penalties. Sáez et al. (2012) propose a hybrid MPC scheme with actuation via holding and stop-skipping using a discrete-time event-based model with stochastic demands. Hernández et al. (2015) improve the work in Delgado et al. (2009, 2012) by extending the MPC-based holding strategy to the case of multiple bus lines. A hybrid MPC design integrating holding, stop-skipping, and short-turning maneuvers is proposed in Nesheli and Ceder (2015) for minimizing passenger delays in a two-way BTS. An MPC formulation based on dynamic running times and demands is developed in Sánchez-Martínez et al. (2016) that involves minimizing passenger delays with holding times as decision variables.

MPC formulations in the bus control literature are based on either non-convex nonlinear programs (NLPs) (Eberlein et al., 2001; Delgado et al., 2009, 2012; Hernández et al., 2015; Sánchez-Martínez et al., 2016) or integer/mixed integer NLPs (Sun and Hickman, 2005; Cortés et al., 2010; Sáez et al., 2012; Nesheli and Ceder, 2015). Computing the global optimum for non-convex NLPs is difficult and is known to be NP-hard (Murty and Kabadi, 1987), whereas it is extremely difficult for integer/mixed integer NLPs which can even be undecidable (Jeroslow, 1973). Thus, with the formulations proposed in the MPC-based holding literature, it is computationally prohibitive to solve the resulting MPC problems to global optimality in real-time, potentially harming the performance of the BTS controller. Furthermore, under conditions of perfect information available to controller, actuation over holding and bus speeds would lead to identical results. However, as uncertainties arising from imperfect information (such as measurement errors and plant/model mismatch) and fast-changing conditions (stochasticity in demands and traffic congestion) are inevitable in practice, it becomes necessary to compensate for them using feedback control. Holding control decisions can be taken only when the bus is stopping at the stop, whereas bus speed control decisions can be applied at every instant of time (except when the bus is stopping at the stop), resulting in the possibility of taking more frequent decisions that are roughly continuous in time with bus speed control methods. Nevertheless, while succeeding in regularizing spacings, the bus speed control methods proposed in the aforementioned studies use formulations that are based only on spacings, which might slow down buses significantly when there are strong variations in the spatial heterogeneity of route congestion.

Building upon earlier conference work in Sirmatel and Geroliminis (2017), this paper extends the bus operations literature in modeling and control aspects: (a) For the modeling part, a novel MLD BTS model is proposed that considers the interaction between bus position and passenger accumulation dynamics. The model is detailed enough to allow in-depth simulation-based analysis of BTSs, but at the same time computationally lightweight to allow for fast execution. (b) In the control aspect, considering a simplified MLD model with actuation via bus speeds, a hybrid MPC scheme is designed 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 exist powerful algorithms for solving the convex QP subproblems (Bonami et al., 2012). In contrast to the works on MPC-based bus control in the literature, the aforementioned feature of the proposed approach enables solutions of the hybrid MPC problems to global optimality while retaining real-time tractability. Furthermore, to the best of the authors' knowledge, this work represents the first attempt on using the MLD modeling approach for BTS modeling and control.

The remainder of the paper is organized as follows: Section 2 introduces the dynamical BTS model based on the MLD modeling framework, intended for use in simulation-based testing of bus operation policies or control schemes. In section 3 a novel hybrid MPC scheme is developed, which is based on a constrained optimal control formulation with the dual objective of spacing regularization and fast BTS operation through coordination of buses by manipulating their speeds in real time. Section 4 contains extensive simulation studies, using the simulation model developed in 2 for representing BTS reality, which showcase the performance of the proposed hybrid MPC scheme. Finally, section 5 provides conclusions and potential directions for future work.

2. Modeling and Simulation of Bus Transport Systems

Using the MLD modeling approach, in this section we develop a novel dynamical BTS model considering the interaction between the dynamics of bus motion and passenger accumulation via formulations integrating physical laws and logic rules. Able to capture detailed dynamical properties of bus operations while also being computationally lightweight, the proposed model can be used for extensive simulation-based analysis of BTS management schemes. In this paper this model is used as the simulation model (i.e., the *plant*) representing the reality of BTS operations for testing the proposed hybrid MPC method, while a simplified version of it will be given as the prediction model of the hybrid MPC scheme later in section 3.

To highlight clearly the contrast between the simulation and prediction models, here we note their important differences: 1) The simulation model has a bus accumulation state representing the number of passengers on a bus with a specific destination stop, while for the bus accumulation state of the prediction model the full destination information is not covered. Instead, the prediction model has two bus accumulation states, one representing the total number of passengers on a bus, whereas the other expresses the part of this total that will alight at the upcoming stop (with some uncertainty). Thus, unlike the prediction model, the simulation model has full information regarding all the destinations of the passengers on a bus. 2) For stops, the simulation model has a stop accumulation state representing the number of passengers at a stop with a specific destination stop, whereas the prediction model has a different stop accumulation state representing only the total number of passengers waiting at a stop without any destination information. Thus, similarly to 1), the simulation model has full information regarding the destinations of all passengers waiting at a stop, whereas the prediction model does not have any information at all regarding their destinations. Hence, the prediction model specifies a simplification over the simulation model given in this section, which is critical for the following reasons: a) The simplification enables computational tractability of the resulting hybrid MPC problems, which is important as the proposed hybrid MPC scheme is intended for real-time bus control. b) It might be difficult to estimate/measure the detailed accumulation states of the simulation model, as they require knowing the destinations of all passengers, complicating usage for control purposes. The prediction model, on the other hand, requires estimation of only the total bus accumulation (which can be easily measured using e.g. optical sensors) and the part alighting at the upcoming stop, which can be extracted (possibly with some errors) from historical data using tap on/tap off systems, whereas the total stop accumulation could be measured with cameras. Thus, owing to its computational benefits and relative ease of the required instrumentation, the prediction model (given later in section 3) specifies a dynamical model suited to real-time control, whereas the simulation model described in this section represents highly detailed BTS dynamics appropriate for in-depth simulation-based analysis of bus operations.

2.1. Bus Line as a Mixed Logical Dynamical System

We consider a single line BTS with K_b buses and K_s stops, and assume that: (i) buses operate on the line with always positive speed (i.e., they never change direction), (ii) a bus always stops at a stop if there are passengers onboard that want to alight at that stop, (iii) passengers do not differentiate between buses when boarding, since any bus they board will stop at their destination stop, (iv) the position of the first stop of the line is assumed to be 0 and the bus positions are reset to 0 when they complete the line and reach the first stop. Furthermore, to facilitate formulation of hybrid dynamics, the dynamical modeling is based on a discrete-time framework, where the system states are updated at each discrete-time step and are thus piecewise constant functions of time.

2.1.1. Dynamics of Continuous States

(a) Dynamics of bus position can be expressed as follows

$$x_i(t+1) = \sum_{j=1}^{K_s} \gamma_{i,j}(t) \left(x_i(t) + Tv_i(t)\right) + \sum_{j=2}^{K_s} \delta_{i,j}(t) x_i(t), \tag{1}$$

for $i = 1, \ldots, K_b$, where t (–) is the time step counter, T (s) is the sampling time, $x_i(t) \in \mathbb{R}$ (m) and $v_i(t) \in \mathbb{R}$ (m/s) are the position (i.e., distance from the first stop) and the speed of bus i, respectively, $\gamma_{i,j}(t) \in \mathbb{B}$ is a binary state expressing whether bus i is cruising to stop j or not, whereas $\delta_{i,j}(t) \in \mathbb{B}$ is a binary state expressing whether bus i is stopping at stop j or not (dynamics of $\gamma_{i,j}(t)$ and $\delta_{i,j}(t)$ are given later). The term $\sum_{j=1}^{K_s} \gamma_{i,j}(t)$ is equal to 1 if bus i is cruising, thus its position will increase. If it is stopping at a stop other than stop 1 (i.e., if $\sum_{j=2}^{K_s} \delta_{i,j}(t)$ is equal to 1) its position will stay constant. If it is stopping at stop 1 (i.e., if $\delta_{i,1}(t) = 1$) this means that it has completed the line and its position will be reset to 0.

(b) Dynamics of bus accumulation can be written as follows

$$n_{i,j}(t+1) = n_{i,j}(t) + T \cdot \left(\sum_{h=1, h \neq j}^{K_s} q^{\text{in}}_{i,h,j}(t) - q^{\text{out}}_{i,j}(t) \right), \tag{2}$$

for $i = 1, \ldots, K_b$ and $j = 1, \ldots, K_s$, where $n_{i,j}(t) \in \mathbb{R}$ (person) is the continuous accumulation of passengers traveling in bus i with destination stop j, $q^{\text{in}}_{i,h,j}(t) \in \mathbb{R}$ (person/s) is the passenger flow with destination stop j boarding bus i at stop h, whereas $q^{\text{out}}_{i,j}(t) \in \mathbb{R}$ (person/s) is the passenger flow alighting from bus i at stop j. Note that we model passenger accumulations and flows as continuous variables instead of integers for simplicity and computational tractability. A bus passenger capacity constraint will be introduced later.

(c) Accumulation at a stop evolves according to the following equation

$$m_{h,j}(t+1) = m_{h,j}(t) + T \cdot \left(\beta_{h,j}(t) - \sum_{i=1}^{K_b} q^{\text{in}}_{i,h,j}(t) \right), \tag{3}$$

for $h, j = 1, \ldots, K_s$, $h \neq j$, where $m_{h,j}(t) \in \mathbb{R}$ (person) is the continuous accumulation of passengers waiting at stop h with destination stop j, whereas $\beta_{h,j}(t) \in \mathbb{R}$ (person/s) is the passenger flow arriving at stop h with destination stop j (i.e., the time varying origin-destination passenger flow demand from stop h to stop j). With each time step the accumulation at stop h will increase with $T\beta_{h,j}(t)$, whereas if there are buses stopping at stop h (i.e., if $\sum_{i=1}^{K_b} q^{\text{in}}_{i,h,j}(t)$ is nonzero) it will decrease as passengers board the buses.

2.1.2. Constraints Defining Binary Events

(a) The event $e^x_{i,j}(t)$ expresses whether bus i has reached stop j ($e^x_{i,j}(t) = 1$) or not $e^x_{i,j}(t) = 0$, and is defined as

$$e^x_{i,j}(t) = \begin{cases} 1 & \text{if } D_j \leq x_i(t) \\ 0 & \text{otherwise,} \end{cases} \tag{4}$$

for $i = 1, \ldots, K_b$, $j = 1, \ldots, K_s$, where $D_j \in \mathbb{R}$ (m) is the distance of stop j from stop 1 (with D_1 specially defined as the length of the whole line).

(b) The event $e^m_h(t)$, storing information on whether there are passengers waiting at stop h ($e^m_h(t) = 0$) or not ($e^m_h(t) = 1$), is defined as

$$e^m_h(t) = \begin{cases} 0 & \text{if } 0 < \sum_{j=1}^{K_s} m_{h,j}(t) \\ 1 & \text{otherwise} \end{cases} \quad \text{for } h = 1, \ldots, K_s. \tag{5}$$

(c) The event $e^c_i(t)$ stores the information on whether bus i is full of passengers ($e_i^c(t) = 1$) or not ($e_i^c(t) = 0$), and is defined as follows:

$$e_i^c(t) = \begin{cases} 0 & \text{if } \sum_{j=1}^{K_s} n_{i,j}(t) < n_{i,\max} \\ 1 & \text{otherwise} \end{cases} \quad \text{for } i = 1, \ldots, K_b, \tag{6}$$

where $n_{i,\max} \in \mathbb{R}$ (person) is the passenger capacity of bus i.

(d) The event $e_{i,j}^n(t)$ expresses whether there are passengers on bus i that want to alight at stop j ($e_{i,j}^n(t) = 0$) or not ($e_{i,j}^n(t) = 1$) and is defined as

$$e_{i,j}^n(t) = \begin{cases} 0 & \text{if } 0 < n_{i,j}(t) \\ 1 & \text{otherwise,} \end{cases} \tag{7}$$

for $i = 1, \ldots, K_b$, $j = 1, \ldots, K_s$.

2.1.3. Dynamics of Binary States

(a) Dynamics of the logical condition expressing whether a bus is cruising to a stop or not can be formulated as follows:

$$\gamma_{i,j}(t+1) = \zeta_{i,j}(t) \vee \eta_{i,j}(t), \tag{8}$$

for $i = 1, \ldots, K_b$, $j = 1, \ldots, K_s$, where $\gamma_{i,j}(t) \in \mathbb{B}$ is the cruising state describing whether bus i is cruising towards stop j ($\gamma_{i,j}(t) = 1$) or not ($\gamma_{i,j}(t) = 0$), $\zeta_{i,j}(t) \in \mathbb{B}$ is the logical condition stating whether bus i begins cruising to stop j ($\zeta_{i,j}(t) = 1$) or not ($\zeta_{i,j}(t) = 0$), whereas $\eta_{i,j}(t) \in \mathbb{B}$ is the logical condition stating whether bus i continues cruising to stop j ($\eta_{i,j}(t) = 1$) or not ($\eta_{i,j}(t) = 0$). The *begin cruising* condition $\zeta_{i,j}(t)$ is defined with a set of events as:

$$\zeta_{i,j}(t) \triangleq \delta_{i,j-1}(t) \wedge \left(\left(e_{j-1}^m(t) \vee e_i^c(t) \right) \wedge e_{i,j-1}^n(t) \right), \tag{9}$$

which can be physically described as follows: Bus i begins cruising to stop j if it is stopping at stop $j-1$ (i.e., if $\delta_{i,j-1}(t)=1$), and there are no passengers wanting to alight at stop $j-1$ (i.e., if $e^n_{i,j-1}(t)=1$), and either there are no passengers at stop $j-1$ wanting to board (i.e., if $e^m_{j-1}(t)=1$) or the bus has no more vacant places (i.e., if $e^c_i(t)=1$). Furthermore, the *continue cruising* condition $\eta_{i,j}(t)$ is defined with a set of events as:

$$\eta_{i,j}(t) \triangleq \gamma_{i,j}(t) \wedge \neg e^x_{i,j}(t), \tag{10}$$

which can be physically described as follows: Bus i continues cruising to stop j if it is cruising (i.e., if $\gamma_{i,j}(t)=1$) and it has not yet reached stop j (i.e., if $e^x_{i,j}(t)=0$). Thus, equation (8) states that bus i begins cruising to stop j if the conditions for it to begin cruising are satisfied (i.e., $\zeta_{i,j}(t)=1$), and it continues cruising to stop j as long as the conditions for its cruising remain satisfied (i.e., $\eta_{i,j}(t)=1$). Once these former conditions are not satisfied anymore, bus i will begin stopping at stop j.

(b) Dynamics of the logical condition expressing whether a bus is stopping at a stop or not can be formulated as follows:

$$\delta_{i,j}(t+1) = \lambda_{i,j}(t) \vee \theta_{i,j}(t), \tag{11}$$

for $i=1,\ldots,K_b$, $j=1,\ldots,K_s$, where $\delta_{i,j}(t)\in\mathbb{B}$ is the stopping state which expresses whether bus i is stopping at stop j ($\delta_{i,j}(t)=1$) or not ($\delta_{i,j}(t)=0$), $\lambda_{i,j}(t)\in\mathbb{B}$ is the logical condition stating whether bus i begins stopping at stop j ($\lambda_{i,j}(t)=1$) or not ($\lambda_{i,j}(t)=0$), whereas $\theta_{i,j}(t)\in\mathbb{B}$ is the logical condition stating whether bus i continues stopping at stop j ($\theta_{i,j}(t)=1$) or not ($\theta_{i,j}(t)=0$). The *begin stopping* condition $\lambda_{i,j}(t)$ is defined as:

$$\lambda_{i,j}(t) \triangleq \gamma_{i,j}(t) \wedge e^x_{i,j}(t), \tag{12}$$

which can be physically described as follows: Bus i begins stopping at stop j if it is cruising to stop j (i.e., if $\gamma_{i,j}(t) = 1$) and it reaches stop j (i.e., if $e^x_{i,j}(t) = 1$). Furthermore, the *continue stopping* condition $\theta_{i,j}(t)$ is defined as:

$$\theta_{i,j}(t) \triangleq \delta_{i,j}(t) \wedge \neg \left(\left(e^m_j(t) \vee e^c_i(t) \right) \wedge e^n_{i,j}(t) \right) , \tag{13}$$

which can be physically described as follows: Bus i continues stopping at stop j if it is stopping (i.e., if $\delta_{i,j}(t) = 1$), and there are passengers wanting to alight (i.e., if $e^n_{i,j}(t) = 0$), or there are passengers wanting to board (i.e., if $e^m_j(t) = 0$) and the bus has vacant places (i.e., if $e^c_i(t) = 0$). Thus, equation (11) states that bus i begins stopping at stop j if the conditions for it to begin stopping are satisfied (i.e., $\lambda_{i,j}(t) = 1$), and it continues stopping at stop j as long as the conditions for its stopping remain satisfied (i.e., $\theta_{i,j}(t) = 1$). Once these former conditions are not satisfied anymore, bus i begins cruising to the next stop, namely $j + 1$.

The simulation model described in this section is a discrete hybrid automaton; its dynamics evolve through interactions of a finite-state machine and a discrete-time linear system. Dynamics of the linear system evolves based on conditions of the binary states (e.g., bus i can move only if $\gamma_{i,j}(t) = 1$ for some j or the passengers can transfer between bus i and stop j only if $\delta_{i,j}(t) = 1$), while the events driving the (binary) state transitions of the finite-state machine are defined by conditions on the continuous states of the linear system. For a clear exposition of the transitions between the binary states, the finite-state machine part of the BTS dynamics is depicted schematically in fig. 1, which is a graphical representation of equations (8) and (11) for bus i and the set of all stops $j \in [1, \ \ldots, \ K_s]$.

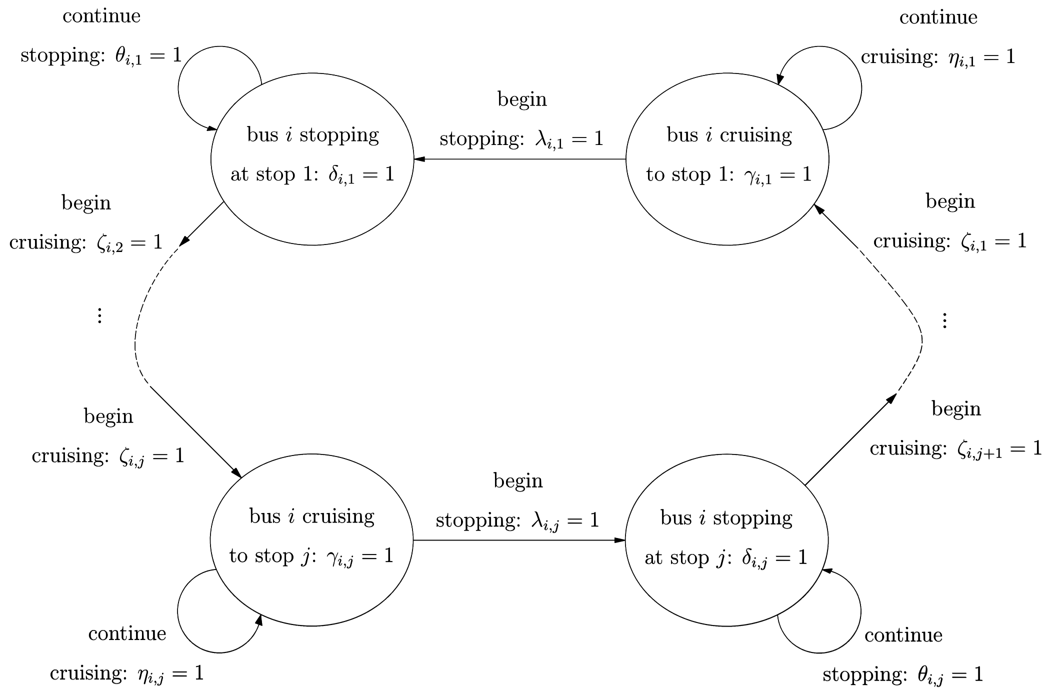

Figure 1: Finite-state machine schematic of the BTS simulation model.

2.1.4. Constraints on Bus Speeds

The speed of bus i is constrained as follows (assuming that the bus is either cruising to or stopping at stop j):

$$v_i(t) = \begin{cases} 0 & \text{if } \delta_{i,j}(t) = 1 \\ \min(\max(u_i(t), v_{\min}), v_{i,\max}(t)) & \text{if } \gamma_{i,j}(t) = 1 \text{ and } u_i(t) \leq \frac{D_j - x_i(t)}{T} \\ \frac{D_j - x_i(t)}{T} & \text{if } \gamma_{i,j}(t) = 1 \text{ and } u_i(t) > \frac{D_j - x_i(t)}{T}, \end{cases} \tag{14}$$

for $i = 1, \ldots, K_b$, where $u_i(t) \in \mathbb{R}$ is the bus speed command calculated by the controller, $v_{\min}$ is a constant value for the minimum bus speed, whereas $v_{i,\max}(t)$ is the maximum bus speed that depends on the traffic conditions of the link on which bus i is currently cruising. The physical reasoning is as follows: (a) If bus i is stopping at time t its speed is 0, (b) if it is cruising at time t and it is far enough that it cannot pass the next stop j during the current time step with $u_i(t)$, then its speed is constrained by the traffic conditions, (c) if it will pass the next stop j (with the speed $u_i(t)$) then its speed is modified such that when the bus reaches the stop its position is equal to the stop position D_j.

2.1.5. Constraints on Passenger Flows

(a) Total passenger flow boarding bus i at stop h cannot exceed the passenger flow capacity $\alpha \in \mathbb{R}$ (person/s) and is allowed to take positive values only when bus i is stopping at stop h, which can be expressed as:

$$\sum_{j=1}^{K_s} q_{i,h,j}^{\text{in}}(t) \leq \alpha \cdot \delta_{i,h}(t), \tag{15}$$

for $i = 1, \ldots, K_b$, $h = 1, \ldots, K_s$.

(b) Total passenger flow boarding bus i at stop h cannot exceed the number of vacant places on bus i, which can be formulated as:

$$T \cdot \sum_{j=1}^{K_s} q_{i,h,j}^{\text{in}}(t) \leq n_{i,\max} - \sum_{j=1}^{K_s} n_{i,j}(t), \tag{16}$$

for $i = 1, \ldots, K_b$, $h = 1, \ldots, K_s$.

(c) Total passenger flow boarding the bus(es) stopping at stop h with destination stop j cannot exceed the number of passengers waiting at stop h with destination stop j, which we express as:

$$T \cdot \sum_{i=1}^{K_b} q_{i,h,j}^{\text{in}}(t) \leq m_{h,j}(t), \tag{17}$$

for $h, j = 1, \ldots, K_s$, $h \neq j$.

(d) Passenger flow alighting from bus i at stop j cannot exceed the passenger flow capacity α and is allowed to take positive values only when bus i is stopping at stop j, which can be expressed as:

$$q_{i,j}^{\text{out}}(t) \leq \alpha \cdot \delta_{i,j}(t), \tag{18}$$

for $i = 1, \ldots, K_b$, $j = 1, \ldots, K_s$.

(e) Passenger flow alighting from bus i at stop j cannot exceed the number of passengers traveling on bus i wanting to alight at stop j, which can be formulated via the following constraint:

$$T \cdot q_{i,j}^{\text{out}}(t) \leq n_{i,j}(t), \tag{19}$$

for $i = 1, \ldots, K_b$, $j = 1, \ldots, K_s$.

2.2. Simulation of Bus Transport Systems

Using the MLD model defined via the dynamical equations (1)-(11), the events (4)-(7), and the constraints (15)-(19), it is possible to simulate the behavior of a single line BTS. The bus speeds $v_i(t)$, passenger flow demands $\beta_{i,j}(t)$, and the initial values of the continuous and binary states are inputs to the simulator. The binary events depend on states and model parameters; thus they can simply be evaluated by considering the corresponding states and the threshold values (e.g., at time t, if $n_{i,j}(t)$ has a nonzero value, then $e_{i,j}^{n}(t)$ is set to 0, otherwise it is set to 1). Decisions of the passengers regarding how they transfer (i.e., $q_{i,h,j}^{\text{in}}(t)$ and $q_{i,j}^{\text{out}}(t)$) depend on the states and model parameters in a more complicated way, which can be modeled using the following linear program (LP):

$$\begin{array}{ll}
\text{maximize} & \sum_{i=1}^{K_b} \sum_{j=1}^{K_s} \left(\sum_{h=1}^{K_s} q^{\text{in}}_{i,h,j}(t) + q^{\text{out}}_{i,j}(t) \right) \\
\text{subject to} & \text{constraints (15)–(19)} \\
& 0 \leq q^{\text{in}}_{i,h,j}(t), \\
& \quad \text{for } i = 1, \dots, K_b,\ h, j = 1, \dots, K_s,\ h \neq j \\
& 0 \leq q^{\text{out}}_{i,j}(t), \\
& \quad \text{for } i = 1, \dots, K_b,\ j = 1, \dots, K_s,
\end{array} \tag{20}$$

where, at time t, the states $\delta_{i,h}(t)$, $n_{i,j}(t)$, and $m_{h,j}(t)$ are given constants. Please note that the LP in equation (20), which is solved at every time instant t (or approximated via suitable heuristics), is simply forcing the passengers to enter in the first available bus that has spare capacity and exit immediately when the bus arrives for the first time in their desired stop. To avoid any confusion, it is not related to optimization-based control of bus operations (i.e., it does not have any direct relation with the hybrid MPC given in section 3), but guarantees a realistic movement of passengers in and out of the bus. More explicitly, the physical interpretation of the problem (20) is that the passengers attempt to transfer at the flow capacity α, but are restricted by physical conditions such as bus capacity constraints, number of people present in buses and at stops, and, by the presence of buses at stops (as modeled via $\delta_{i,h}(t)$). The procedure for simulating the BTS is given in algorithm 1.

3. Control of Bus Transport Systems

Achieving schedule reliability is an important goal in BTS control. This corresponds to minimizing the standard deviation of headways, as headways

Algorithm 1 MLD bus line simulation

1) Initialize simulation at $t = 1$ with the initial states $x_i(1)$, $n_{i,j}(1)$, $m_{h,j}(1)$, $\gamma_{i,j}(1)$, and $\delta_{i,j}(1)$.
2) At time step t, given the previous states $x_i(t)$, $n_{i,j}(t)$, $m_{h,j}(t)$, $\gamma_{i,j}(t)$, and $\delta_{i,j}(t)$, evaluate $e^x_{i,j}(t)$, $e^m_j(t)$, $e^c_i(t)$, and $e^n_{i,j}(t)$, and calculate $q^{\text{in}}_{i,h,j}(t)$ and $q^{\text{out}}_{i,j}(t)$ by solving the optimization problem (20) (or, via heuristics).
3) Given the values obtained at step 2, the previous states, and the exogenous inputs $v_i(t)$ and $\beta_{h,j}(t)$, evolve system dynamics by evaluating the difference equations (1)-(11) to obtain the successor states $x_i(t+1)$, $n_{i,j}(t+1)$, $m_{h,j}(t+1)$, $\gamma_{i,j}(t+1)$, and $\delta_{i,j}(t+1)$.
4) Repeat steps 2 and 3 for $t = 1, \ldots, t_{\text{final}}$.

are related to 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 in the control aspect: A novel hybrid MPC scheme that optimizes future BTS trajectories (predicted using a prediction model that specifies a simplified version of the fully detailed MLD model described in section 2) 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.

3.1. Hybrid Model Predictive Controller

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. The prediction model of the hybrid MPC includes all K_b buses operating on the line 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 version of the BTS simulation model given in section 2 involving a finite horizon in both time and space: 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. A prediction model based on the complete BTS dynamics given in section 2 would result in intractable hybrid MPC problems due to excessive computational burden associated with the large number of binary variables, whereas the simplified version given here yields a computationally efficient formulation required for real-time control.

3.1.1. Dynamics of Continuous States (Prediction Model)

(a) Predicted bus position dynamics can be written as (analogous to equation (1)):

$$x_i(k+1) = x_i(k) + T \cdot z_i(k), \tag{21}$$

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 (2)):

$$\begin{aligned} n_i(k+1) &= n_i(k) + T \cdot \left(q_i^{\text{in}}(k) - q_i^{\text{out}}(k)\right) \\ n_{i,a}(k+1) &= n_{i,a}(k) - T q_i^{\text{out}}(k), \end{aligned} \tag{22}$$

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. Equation (22) is a simplified version of equation (2), where the former has two separate states both with only one index since in the prediction model only the upcoming stop is included, whereas the latter includes the bus accumulation states for all bus-stop pairs in the line.

(c) Dynamics of the predicted stop accumulation state $m_a(k)$ can be written as follows (analogous to equation (3)):

$$m_a(k+1) = m_a(k) + T \cdot \left(\beta_a(k, t_c) - \sum_{i=1}^{\mathcal{I}^a} q_i^{\text{in}}(k) \right). \tag{23}$$

Equation (23) is a simplification of equation (3), where the former has only one index since in the prediction model the bus can only interact with its upcoming stop regarding passenger transfers (and the rest of the stops are omitted), whereas in the latter the stop accumulation states for all origin-destination pairs are defined. Here, $\beta_a(k, t_c) \in \mathbb{R}$ 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.

3.1.2. Constraints Defining Binary Events (Prediction Model)

(a) The event $e_i^x(k) \in \mathbb{B}$ expresses (analogous to equation (4)) whether bus i has reached stop a or not, and is defined as:

$$e_i^x(k) = \begin{cases} 1 & \text{if } 0 \leq x_i(k) \\ 0 & \text{otherwise.} \end{cases} \tag{24}$$

(b) The event $e_i^m(k) \in \mathbb{B}$ describes (analogous to equation (5)) whether there are any passengers at stop a or not:

$$e_i^m(k) = \begin{cases} 1 & \text{if } m_a(k) \leq 0 \\ 0 & \text{otherwise.} \end{cases} \tag{25}$$

(c) The event $e_i^c(k) \in \mathbb{B}$ expresses (analogous to equation (6)) whether bus i has available space for passengers or not:

$$e_i^c(k) = \begin{cases} 1 & \text{if } n_{i,\max} - n_i(k) \leq 0 \\ 0 & \text{otherwise.} \end{cases} \tag{26}$$

(d) The event $e_i^n(k) \in \mathbb{B}$ describes (analogous to equation (7)) whether there are passengers on bus i that want to alight at stop a or not:

$$e_i^n(k) = \begin{cases} 1 & \text{if } n_{i,a}(k) \leq 0 \\ 0 & \text{otherwise.} \end{cases} \tag{27}$$

3.1.3. Dynamics of Binary States (Prediction Model)

(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:

$$\gamma_i(k+1) = \gamma_i(k) \wedge \neg e_i^x(k). \tag{28}$$

Equation (28) specifies a simplification over equation (8), where the former has only one index as the bus can only cruise to its upcoming stop (and then stop there) while the rest of the stops are not included in the prediction model, whereas in the latter the cruising states for all bus-stop pairs are defined since all stops are taken into account in the simulation model given in section 2. Furthermore, there is no *begin cruising* condition in equation (28) (in contrast to (8)), since in the prediction model a bus is simply assumed to be either cruising to or stopping at its upcoming stop, without any previous binary states from which to transition.

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

$$\delta_i(k+1) = \lambda_i(k) \vee \theta_i(k), \tag{29}$$

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 ($\theta_i(k) = 0$). Equation (29) is a simplified version of equation (11), where the former has only one index as the bus can only stop at its upcoming stop (and then leave) while the rest of the stops are omitted from the prediction model, whereas in the latter the stopping states for all bus-stop pairs are defined. The predicted *begin stopping* condition $\lambda_i(k)$ is defined as (analogous to equation (12)):

$$\lambda_i(k) \triangleq \gamma_i(k) \wedge e_i^x(k), \tag{30}$$

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 (analogous to (13)):

$$\theta_i(k) \triangleq \delta_i(k) \wedge \neg\big((e_i^m(k) \vee e_i^c(k)) \wedge e_i^n(k)\big), \tag{31}$$

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 (29) 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 (in contrast to the simulation model) since the stops other than the upcoming stop are not included in the prediction model.

Similar to the simulation model given in section 2, the prediction model described here is also a discrete hybrid automaton, although with a special structure of the finite-state machine. Owing to the fact that only the upcoming stop for each bus is included in the model (arising from the finite horizon in space approximation employed for computational efficiency reasons), the state transition structure of the binary state dynamics forms an open chain (in contrast to the closed chain in fig. 1), having only two binary states. This open chain structure is related to the scenario assumed for the duration of the prediction horizon: In this scenario, at the first step of the prediction horizon, bus i is either cruising to its upcoming stop a or stopping there. During the period when it is stopping at a passengers transfer, and then the stopping state becomes inactive, allowing the bus to move. As only the upcoming stop is considered with a finite time horizon, there is no transition from the stopping state δ_i to further cruising states for stops beyond a. To draw clearly the contrast between the simulation model given in section 2 and the prediction model, the finite-state machine part for the prediction model is depicted schematically in fig. 2, which is a graphical representation of equations (28) and (29) for bus i and its upcoming stop a.

3.1.4. Constraints on Bus Speed Control Inputs (Prediction Model)

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

$$z_i(k) = \begin{cases} u_i(k) & \text{if } \delta_i(k) = 0 \\ 0 & \text{otherwise,} \end{cases} \tag{32}$$

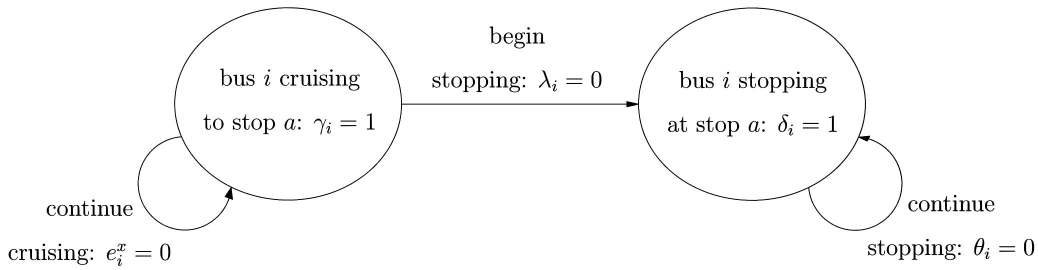

Figure 2: Finite-state machine schematic of the BTS prediction model.

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:

$$v_{\min} \leq u_i(k) \leq v_{a,\max}(k, t_c), \tag{33}$$

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.

3.1.5. Constraints on Passenger Flows (Prediction Model)

(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 (analogous to equations (15), (16), and (17)):

$$\begin{aligned} q_i^{\text{in}}(k) &\leq \alpha \cdot \delta_i(k) \\ T \cdot q_i^{\text{in}}(k) &\leq n_{i,\max} - n_i(k) \\ T \cdot q_i^{\text{in}}(k) &\leq m_a(k). \end{aligned} \tag{34}$$

(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 (analogous to equations (18) and (19)):

$$q_i^{\text{out}}(k) \leq \alpha \cdot \delta_i(k) \tag{35}$$
$$T \cdot q_i^{\text{out}}(k) \leq n_i(k).$$

3.1.6. Initial States (Prediction Model)

Initial bus position state is defined as:

$$x_i(1) = -d_{i,a}(t_c), \tag{36}$$

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:

$$\gamma_i(1) = \tilde{\gamma}_{i,a}(t_c), \tag{37}$$

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

$$\delta_i(1) = \tilde{\delta}_{i,a}(t_c), \tag{38}$$

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

$$m_a(1) = \sum_{j=1}^{K_s} \tilde{m}_{a,j}(t_c), \tag{39}$$

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

$$n_i(1) = \sum_{j=1}^{K_s} \tilde{n}_{i,j}(t_c) \tag{40}$$
$$n_{i,a}(1) = \tilde{n}_{i,a}(t_c),$$

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.

3.1.7. 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:

$$\begin{aligned}
\underset{u_i(k)}{\text{minimize}} \quad & \sum_{i=1}^{K_b}\sum_{k=1}^{N}\left(e_i^2(k+1)+\sigma\cdot y_i^2(k)\right) \\
\text{subject to} \quad & \text{for } i=1,\ \ldots,\ K_b: \\
& \quad \text{initial states (37), (38), (39), (40)} \\
& \quad \text{for } k=1,\ \ldots,\ N: \\
& \qquad 0\le q_i^{\text{in}}(k) \\
& \qquad 0\le q_i^{\text{out}}(k) \\
& \qquad \text{dynamics (21), (28), (29), (23), (22)} \\
& \qquad \text{constraints (32), (33), (34), (35)} \\
& \qquad \text{events (24), (25), (26), (27),}
\end{aligned} \tag{41}$$

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:

$$\begin{aligned}
e_i(k) &= x_{i,f}(k)-2x_i(k)+x_{i,r}(k) \\
y_i(k) &= v_{a,\max}(k,t_c)-z_i(k),
\end{aligned} \tag{42}$$

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 (41) 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.

3.2. Integral and Proportional-Integral Controllers

We first define front and rear spacings as follows:

$$s_{i,f}(t) = x_{i,f}(t) - x_i(t) \tag{43}$$

$$s_{i,r}(t) = x_i(t) - x_{i,r}(t), \tag{44}$$

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:

$$e_i(t) = s_{i,f}(t) - s_{i,r}(t). \tag{45}$$

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:

$$u_i(t) = u_i(t-1) + K_{\mathrm{I}} \cdot e_i(t), \tag{46}$$

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:

$$u_i(t) = u_i(t-1) + K_{\mathrm{P}} \cdot (e_i(t) - e_i(t-1)) + K_{\mathrm{I}} \cdot e_i(t), \tag{47}$$

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.

4. Simulation Results

4.1. Results of a Congested Scenario

A bus line with 8 buses and 32 stops is considered, where each link is 1 km long. 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 $t_{\mathrm{final}} = 6480$ 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 passengers for all buses, 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 $K_{\mathrm{P}} = 1.04$ and $K_{\mathrm{I}} = 0.146$, while weighting factor of the hybrid MPC scheme is chosen as $\sigma = 7000$ (reasoning for these will be given later). 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, using algorithm 1 for simulating BTS dynamics (i.e., to represent reality). Note that the execution time of the simulation algorithm 1 itself is negligible as it takes around 1 milliseconds to execute one time step (thus, for the present scenario with $t_{\text{final}} = 6480$ steps, the whole simulation takes around 7 seconds), indicating its computationally lightweight nature. Sensitivity analysis on the bus passenger capacity is also provided later as it provides interesting observations about the operations of the system.

Simulation results with the congested scenario are given in figs. 3 to 6, whereas a summary with numerical values of various performance criteria is given in table 1. In fig. 3 the bus positions (i.e., time-space diagrams) are shown for a period of simulation time (that would correspond to the first 4 hours of real time) for the three proposed controllers, where each different color represents one bus. From fig. 3 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 4 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

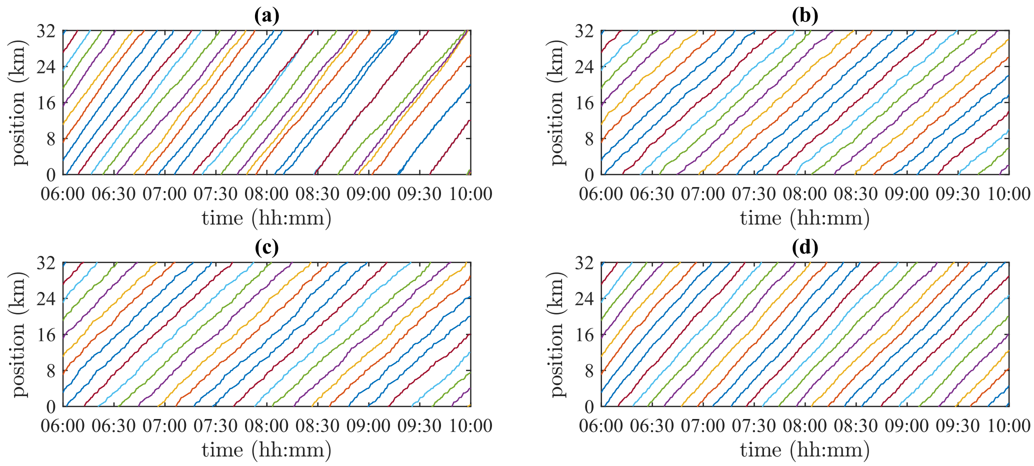

Figure 3: Bus positions for the first 4 hours of the congested scenario: (a) no control, (b) I-controller, (c) PI-controller, (d) hybrid MPC.

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 and around the peak hours, ultimately yielding decreased service times.

Figure 5 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

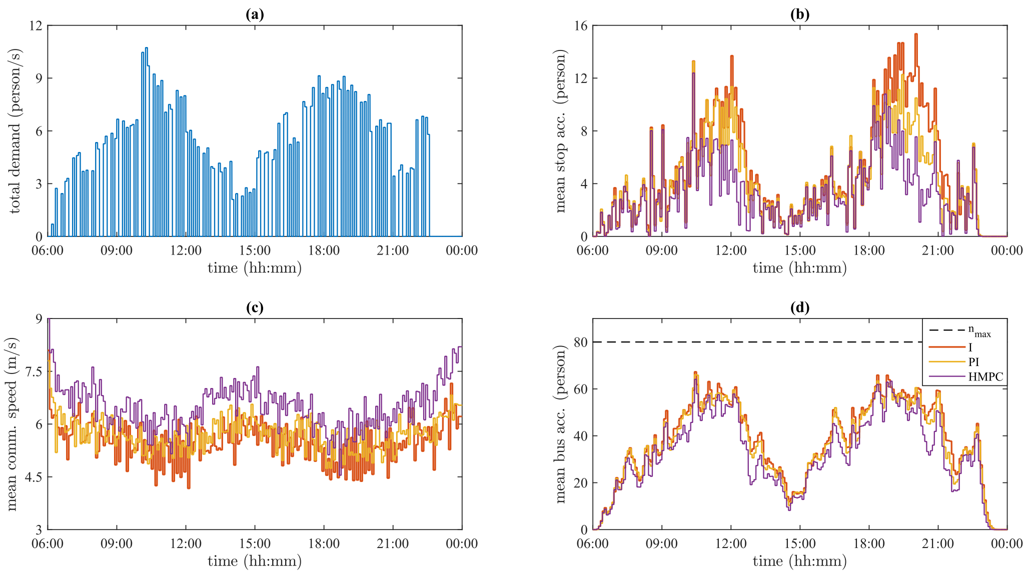

Figure 4: Total demand (a), mean stop accumulation (b), mean commercial speed (c), and mean bus accumulation (d), as functions of time, for the congested scenario.

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 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 6 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.

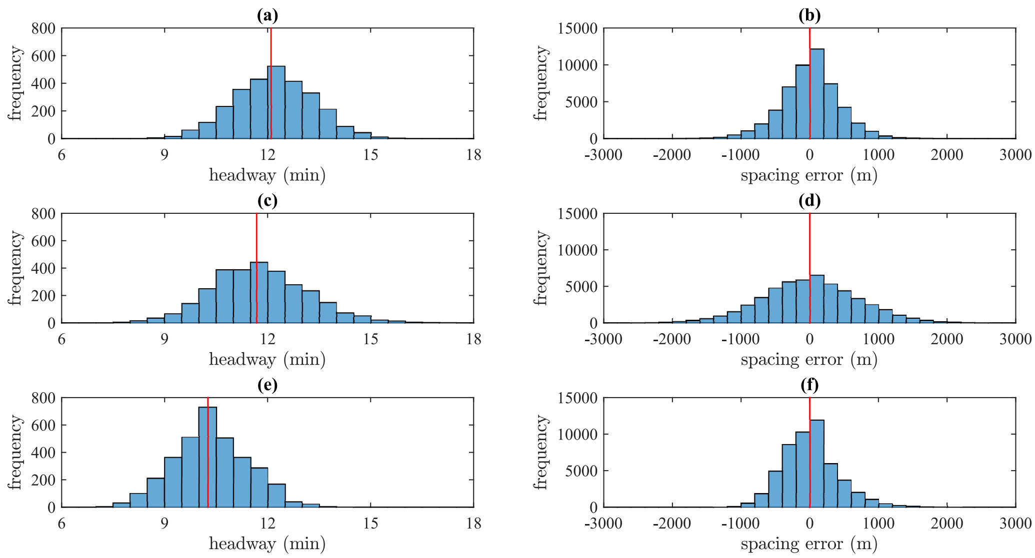

Figure 5: Headway ((a), (c), (e)) and spacing error ((b), (d), (f)) distributions of the congested scenario (vertical red lines show the mean values, which are 0 for spacing errors by definition): (a)-(b) I-controller, (c)-(d) PI-controller, (e)-(f) hybrid MPC.

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

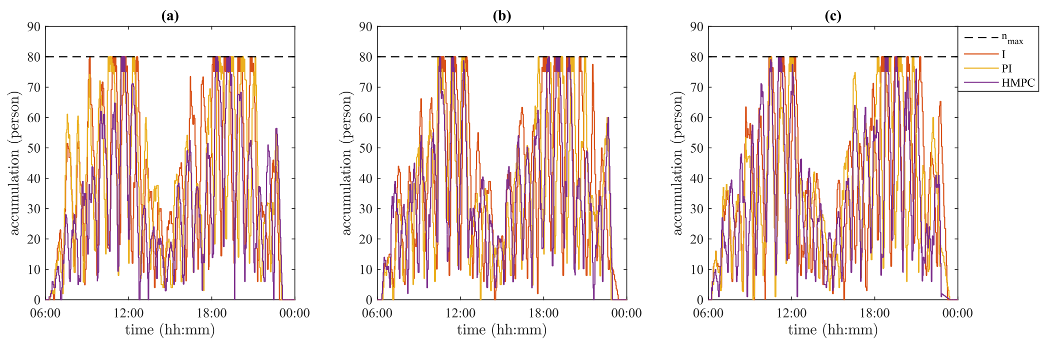

Figure 6: Accumulations of (a) bus 1, (b) bus 2, and (c) bus 3, for I- and PI-controllers and the hybrid MPC for the congested scenario.

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 higher mean commercial 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. 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 21% compared to the I- and 15% compared to the PI-controller, (b) it improves headway regularity by 5% compared to the I- and 23% 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 0.35/0.7 s needed for solving one instance of the MIQP given in (41) 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.

4.2. 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 more general 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. 7, 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 18% against the I- and 14% against the PI-controller, at the same time showing an improvement in headway regularity around 9% against the I- and 28% 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

Table 1: Control Performance Evaluation for a Congested Scenario

performance criterion	NC	I	PI	HMPC
mean time spent at stop (min)	19.5	9.5	8.2	6.2
mean time spent in bus (min)	13.4	18.6	17.9	15.9
mean commercial speed (m/s)	7.7	5.5	5.7	6.5
mean of headways (min)	–	12.1	11.7	10.3
standard deviation of headways (min)	–	1.12	1.38	1.07
standard deviation of spacing errors (m)	–	409	690	400
mean/max computation time (s)	–	–	–	0.35/0.7

large intervals (roughly between 24 and 28.5 minutes for the I- and 24 and 27.5 minutes for the PI-controller), the hybrid MPC consistently performs in the 21-22.5 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, 17% faster than the I- and 15% faster than the PI-controller) than the other controllers. Performance of the hybrid MPC in spacing regularity is more modest compared to that of headways, as seen from the spacing regularity figure, although it is still superior to the I- and PI- controllers. 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). All these three 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.

4.3. Controller Tuning

Performance of the proposed controllers depend on the controller gains for the I- and PI-controllers, whereas for HMPC it is a function of the objective function weight σ. A series of simulation experiments with the congested scenario setup presented in the previous section are conducted to evaluate how the BTS performance changes with the controller parameters, where a set of 100 randomly generated scenarios is repeated for each value of K_I, K_P, and

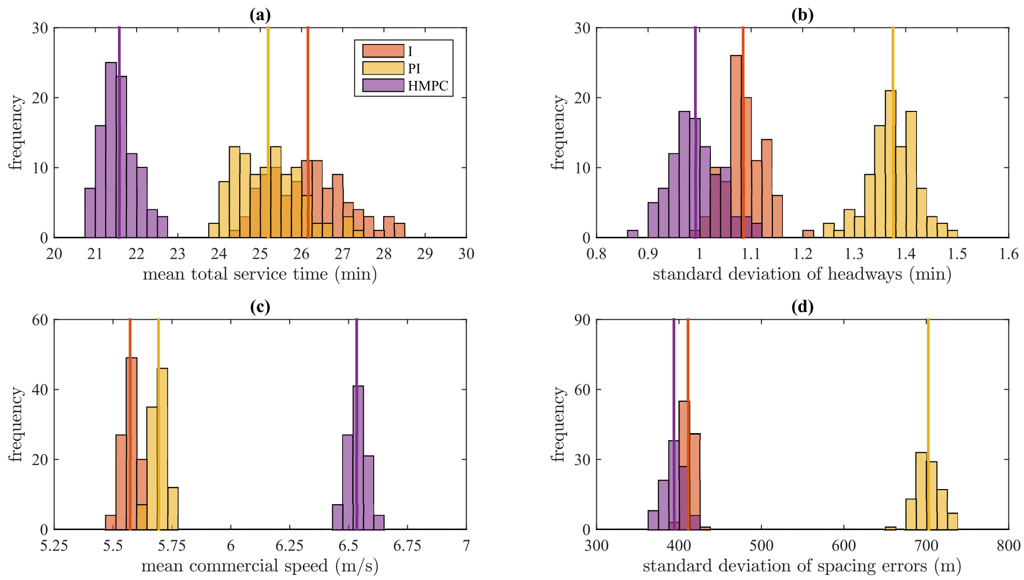

Figure 7: Histograms of the performance criteria from 100 randomly generated scenarios, comparing the I- and PI- controllers with the hybrid MPC: (a) mean total service time, (b) standard deviation of headways, (c) mean commercial speed, (d) standard deviation of spacing errors.

σ. Since PI-controller has two parameters to tune, its K_I parameter is taken as fixed to the best found from the tuning of the I-controller. The results are shown in fig. 8, which includes plots of mean TSS, mean TSB, standard deviation of headways, mean commercial speed, and normalized mean objective (i.e., the total of mean TSS, mean TSB, and standard deviation of headways, separately for each controller, normalized between 0 and 1) as functions of the controller parameters. These results indicate that: (a) Overall performance of the I-controller improves with increasing K_I up to a point, and then there is no change, which can be explained by the fact that it is only reacting to the spacing error and thus has no trade-off. The PI-controller reacts to both spacing error (related to headways) and its rate of change (related to speed), and thus yields a trade-off between service times/speeds and headway regularity, which can be clearly seen from the figure where mean TSS and TSB decrease (along with increasing mean commercial speed) while standard deviation of headways increases, with increasing K_P. Although its overall performance is superior, tuning results of hybrid MPC suggest a trade-off very similar to that of the PI-controller, which is to be expected due to its objective function having two terms related to spacing errors and speeds. The controller parameter values that attain a minimum for the normalized mean objective curves (i.e., $K_{\mathrm{I}} = 0.146$, $K_{\mathrm{P}} = 1.04$, and $\sigma = 7000$) indicate a balance between fast operation (i.e., small TST) and headway regularity, and thus are chosen for the results presented in the previous section with the congested scenario.

4.4. Sensitivity to Bus Capacity

Bus capacity is critical as it has a strong effect on the overall behavior of the BTS. Moreover, it is interesting to explore if the BTS can be operated with buses with smaller capacities without harming performance. This can create significant savings in bus operations, given that many companies have fleets of different types. To analyze how changing the bus capacity affects the BTS performance, a series of simulation experiments are conducted comparing the proposed controllers with the simulation setup in the congested scenario together with bus capacity values changing from 50 to 100. The results are given in fig. 9, which includes plots showing the four performance criteria as a function of bus capacity for the three controllers. From these

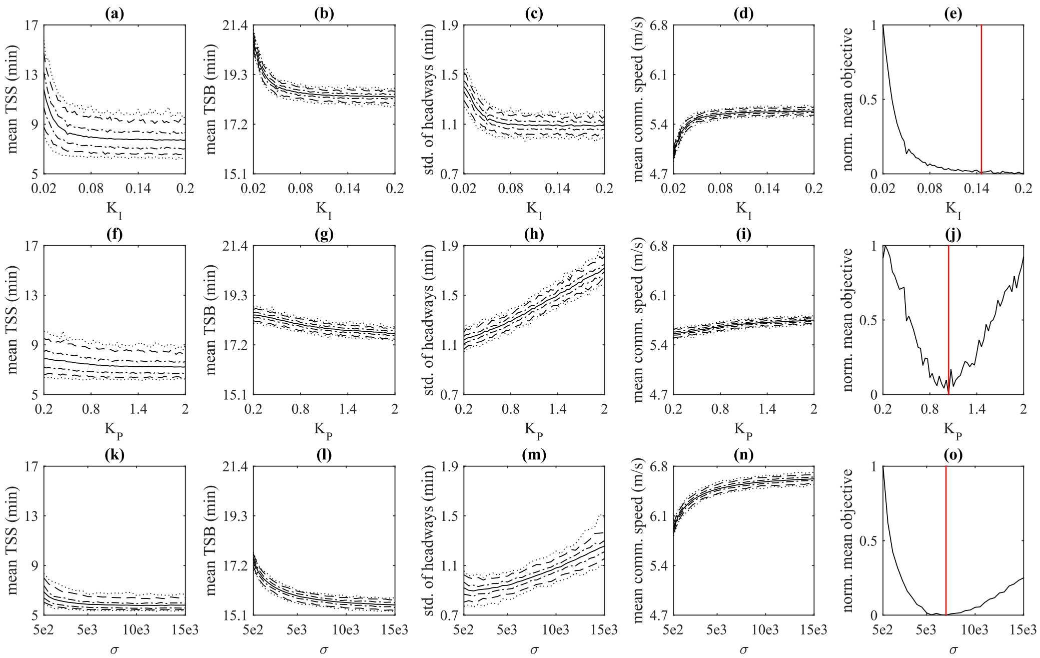

Figure 8: Results of controller tuning showing the mean together with the 1st, 5th, 25th, 75th, 95th, and 99th percentiles of the mean time spent at stop (TSS) ((a), (f), (k)), mean time spent in bus (TSB) ((b), (g), (l)), standard deviation of headways ((c), (h), (m)), mean commercial speed ((d), (i), (n)), and normalized mean objective ((e), (j), (o)) (with its minimum indicated by a vertical red line) for I-controller (a)-(e), PI-controller (f)-(j), and hybrid MPC (k)-(o), as functions of K_I, K_P, and σ, respectively.

results, two observations can be made: (1) Performance of hybrid MPC is insensitive to changes in the bus capacity values in the 65-100 range, whereas I- and PI-controllers suffer from increased mean TSS values especially for $n_{\max} \leq 70$, (2) Performance criteria other than the mean TSS seem to be insensitive to changes in bus capacity. We can thus conclude that by using a feedback controlled BTS (especially with a hybrid MPC), it is possible to use smaller buses without harming the BTS performance, indicating poten-

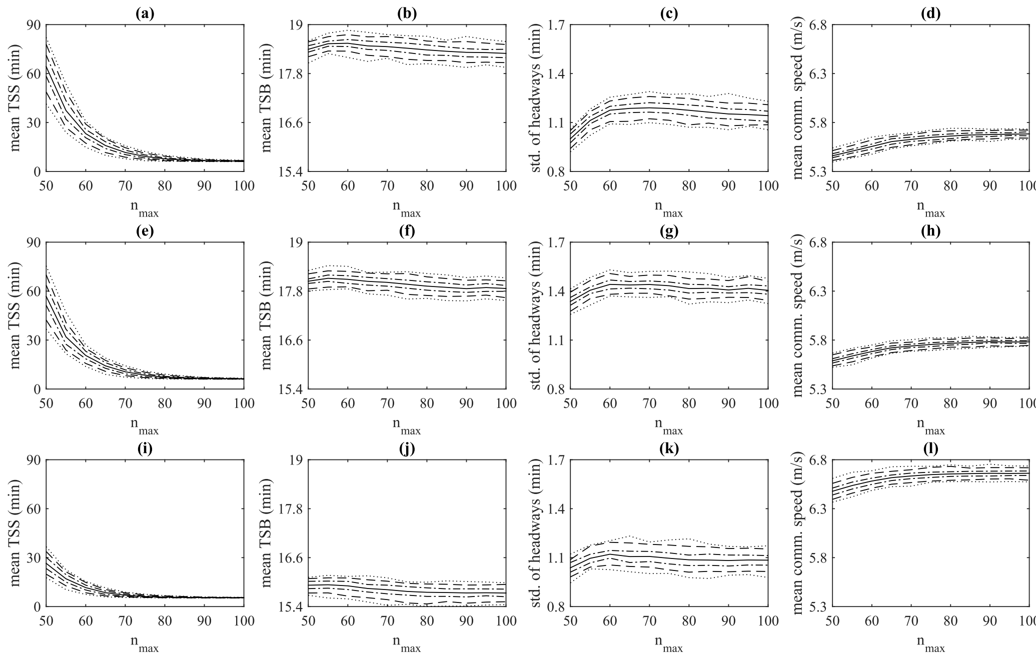

Figure 9: Results of sensitivity analysis of BTS performance to bus capacity values showing the mean together with the 1st, 5th, 25th, 75th, 95th, and 99th percentiles of the mean time spent at stop (TSS) ((a), (e), (i)), mean time spent in bus (TSB) ((b), (f), (j)), standard deviation of headways ((c), (g), (k)), mean commercial speed ((d), (h), (l)), for I-controller (a)-(d), PI-controller (e)-(h), and hybrid MPC (i)-(l), as functions of $n_{\max}$.

tials in cost savings due to decreased bus sizes. Future work can further investigate the inconvenience caused by overcrowding of smaller buses and possibly the effect of dwell times, which in our study are considered constant and not influenced by overcrowding. While such an addition is straightforward in terms of formulation (e.g., the relationship between dwell times and overcrowding can be expressed using a piecewise affine function), real data is required for calibration purposes.

5. Conclusion

We developed a computationally efficient MLD BTS model able to capture detailed dynamics of single line bus operations, involving interactions of bus motion and passenger flows between buses and stops. Taking 1 milliseconds per iteration, this BTS model can be used for simulation-based in-depth analyses of bus networks for evaluation of bus management/control schemes. Furthermore, we proposed a novel hybrid MPC design based on a simplified 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 line 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 real-time feasible 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 simulations using the proposed MLD model. Results indicate the potential of the hybrid MPC in decreasing service times and improving headway regularity, even for decreased bus capacity values. 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-line bus systems, establishing stability properties of the controllers, and comparisons of the proposed model with microscopic BTS simulation frameworks.

References

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

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

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

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

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

Berrebi, S. J., Watkins, K. E., Laval, J. A., 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., Kilinç, M., Linderoth, J., 2012. Algorithms and software for convex mixed integer nonlinear programs. In: Mixed Integer Nonlinear Programming. Springer, pp. 1–39.

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

Chow, A. H., Li, S., Zhong, R., 2017. Multi-objective optimal control formulations for bus service reliability with traffic signals. Transportation Research Part B: Methodological 103, 248–268.

Cortés, C. E., Sáez, D., Milla, F., Núñez, A., Riquelme, M., 2010. Hybrid predictive control for real-time optimization of public transport systems' operations based on evolutionary multi-objective optimization. Transportation Research Part C: Emerging Technologies 18 (5), 757–769.

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

Delgado, F., Contreras, N., Munoz, J. C., 2013. Holding for transfers. In: Transportation Research Board 92nd Annual Meeting. No. 13-2913.

Delgado, F., Muñoz, J., Giesen, R., Cipriano, A., 2009. Real-time control of buses in a transit corridor based on vehicle holding and boarding limits. Transportation Research Record: Journal of the Transportation Research Board (2090), 59–67.

Delgado, F., Muñoz, J. C., Giesen, R., 2012. How much can holding and/or limiting boarding improve transit performance? Transportation Research Part B: Methodological 46 (9), 1202–1217.

Eberlein, X. J., Wilson, N. H., Bernstein, D., 2001. The holding problem with real–time information available. Transportation science 35 (1), 1–18.

Fu, L., Liu, Q., Calamai, P., 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., Prett, D. M., Morari, M., 1989. Model predictive control: theory and practice – a survey. Automatica 25 (3), 335–348.

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

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

Hans, E., Chiabaut, N., Leclercq, L., 2015a. Investigating the irregularity of bus routes: highlighting how underlying assumptions of bus models impact the regularity results. Journal of Advanced Transportation 49 (3), 358–370.

Hans, E., Chiabaut, N., Leclercq, L., Bertini, R. L., 2015b. Real-time bus route state forecasting using particle filter and mesoscopic modeling. Transportation Research Part C: Emerging Technologies 61, 121–140.

Hernández, D., Muñoz, J. C., Giesen, R., Delgado, F., 2015. Analysis of real-time control strategies in a corridor with multiple bus services. Transportation Research Part B: Methodological 78, 83–105.

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

Jeroslow, R., 1973. There cannot be any algorithm for integer programming with quadratic constraints. Operations Research 21 (1), 221–224.

Liu, H., Skabardonis, A., Zhang, W.-b., 2003. A dynamic model for adaptive bus signal priority. In: Preprint CD-ROM, 82nd Transportation Research Board Annual Meeting, Washington, DC.

Löfberg, J., 2004. YALMIP: A toolbox for modeling and optimization in MATLAB. In: IEEE International Symposium on Computer Aided Control Systems Design. IEEE, pp. 284–289.

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

Murty, K. G., Kabadi, S. N., 1987. Some np-complete problems in quadratic and nonlinear programming. Mathematical programming 39 (2), 117–129.

Nesheli, M. M., Ceder, A. A., 2015. A robust, tactic-based, real-time framework for public-transport transfer synchronization. Transportation Research Part C: Emerging Technologies 60, 105–123.

Newell, G. F., Potts, R. B., 1964. Maintaining a bus schedule. In: Australian Road Research Board (ARRB) Conference, 2nd, 1964, Melbourne. Vol. 2. p. 388393.

Sáez, D., Cortés, C. E., Milla, F., Núñez, A., Tirachini, A., Riquelme, M., 2012. Hybrid predictive control strategy for a public transport system with uncertain demand. Transportmetrica 8 (1), 61–86.

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

Schmöcker, J.-D., Sun, W., Fonzone, A., Liu, R., 2016. Bus bunching along a corridor served by two lines. Transportation Research Part B: Methodological 93, 300–317.

Sirmatel, I., Geroliminis, N., 2017. Dynamical modeling and predictive control of bus transport systems: A hybrid systems approach. Vol. 50. pp. 7499 – 7504, 20th IFAC World Congress.

Sun, A., Hickman, M., 2005. The real–time stop–skipping problem. Journal of Intelligent Transportation Systems 9 (2), 91–109.

Van Der Schaft, A. J., Schumacher, J. M., 2000. An introduction to hybrid dynamical systems. Vol. 251. Springer London.

Van Oort, N., Boterman, J., Van Nes, R., 2012. The impact of scheduling on service reliability: trip-time determination and holding points in long-headway services. Public Transport 4 (1), 39–56.

Wu, W., Liu, R., Jin, W., 2017. Modelling bus bunching and holding control with vehicle overtaking and distributed passenger boarding behaviour. Transportation Research Part B: Methodological 104, 175–197.

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