2018 European Control Conference (ECC)
June 12-15, 2018. Limassol, Cyprus

Hybrid Model Predictive Control of Bus Transport Systems

Isik Ilber Sirmatel and Nikolas Geroliminis

Abstract— **Schedule instability adversely affects the performance of bus transport systems. Bus bunching and similar irregularities decrease quality of bus services and increase travel times. In this regard, bus system management schemes are of high importance. Motivated by the potential impact of developing advanced bus control schemes on transportation practice, in this paper a hybrid model predictive control scheme with actuation via bus speeds is developed, which can regularize headways and improve bus service quality. The controller extends upon earlier work by considering detailed dynamics of the interactions between buses and stops via passenger flows. Performance of the controller is compared with a no control case and a PI controller via simulation experiments, conducted with a recently proposed mixed logical dynamical bus loop model involving both continuous (e.g., bus positions) and binary (e.g., the state of a bus regarding whether it is holding at a certain stop or not) states. Results showcase the capability of the proposed controller in regularizing headways, avoiding bus bunching and decreasing passenger travel times.**

I. INTRODUCTION

Bus operations are known to be inefficient when operated without control [1]. Buses that are slow to arrive at the stop encounter more passengers waiting to board, delaying those buses even more, while faster buses encounter less passengers. This positive feedback gives rise to bus bunching, a highly undesirable phenomenon in bus transport system (BTS) operations. Spatiotemporal variabilities in traffic conditions and demands lead to irregularities in headways and ultimately to bus bunching, causing inefficient operations and poor service quality. Research on developing models and control schemes for BTSs is thus of high importance. Especially in the last decades, many researchers focused on developing methods for real-time bus control, to avoid bus bunching and ensure efficient BTS operations. Bus control methods are classified into two categories: 1) Station control, 2) inter-station control (see [2] for an extensive review, a summary of which is given in this section).

Station control methods involve control input updates at a subset of stops. Some of these methods focus on holding for headway regularization, assuming this would improve operations and decrease travel times [3], [4], [5]. Passenger delays need to be considered in addition to headways when there is high variability in the demands [6], [7], [2]. Holding can also be used to improve timing of passenger transfers [8], [9]. Stop-skipping strategies specify another class of station control methods. Here, to increase speed and efficiency, buses are forced to skip stops when needed and possible [10], [11], [7]. Showing potential in headway regularization for medium intensity demands, station control methods might decrease BTS service quality as their actuation is via holding the bus at a stop or forcing the bus skip a stop. The buses may be forced to operate at lower speeds under some conditions, leading not only to degradations in BTS performance, but also increases in costs and required fleet sizes.

The authors are with the School of Architecture, Civil and Environmental Engineering, École Polytechnique Fédérale de Lausanne (EPFL), 1015 Lausanne, Switzerland. {isik.sirmatel,nikolas.geroliminis}@epfl.ch

Inter-station control methods constitute another category of bus system control. Here control decisions are taken while the bus is cruising. Traffic signal priority, a class of inter-station control, involves traffic flow manipulation by giving priority to certain circulations to favor bus system operation [12], [13]. Methods involving bus speed actuation also belong to the inter-station control strategies. With bus speed actuation, the controller can manipulate bus speeds for bus bunching avoidance and improved performance. On the speed control direction, a mixed bus speed control and signal priority method is developed in [14], which can equalize headways at a desired value. In [1] a spacing based bus speed controller is designed, which calculates the control input based on the front and rear spacings. The method is able to satisfy bounds on bus speeds and prevent bus bunching. A combined estimation and control scheme is proposed in [15] to achieve BTS coordination for headway regularization. Through predictions on passenger accumulations waiting to board at each stop and those travelling on the buses, it is possible to improve BTS performance and regularize headways under congested conditions.

Motivated by need for developing advanced bus control methods to increase BTS efficiency, in this paper we propose a hybrid model predictive control (MPC) scheme able to achieve headway regularity and fast BTS operation. The scheme can be classified as a bus speed controller under the inter-station control category, and combines bus motion dynamics with the interactions between buses and stops via passenger flows in its prediction model. The model is based on the the mixed logical dynamical (MLD) modeling framework proposed by [16] for modeling dynamical systems involving the interaction of physical laws, logic rules, and constraints. Performance of the controller is compared with a no control case and a PI controller using a detailed BTS model recently proposed in [17]. The BTS model captures dynamics of bus motion and passenger accumulations both at the stops and in the buses, and also includes interactions between stops and buses via passenger flows. The model enables computationally lightweight but in-depth simulation-based testing of BTS control methods. Results indicate the high potential of the proposed hybrid MPC scheme in headway regularization and decreasing passenger delays.

978-3-9524-2699-9 ©2018 EUCA

II. MODELING

A. Mixed Logical Dynamical Model of a Bus Loop

We consider a bus loop with K_b buses and K_s stops, with the assumptions that (i) buses always have positive speed (i.e., no changes in direction), (ii) a bus is required to hold at a stop if there are passengers desiring to alight at that stop, (iii) passengers simply board any bus arriving at a stop, since any bus they board will travel to their destination stop (as it is a single loop), (iv) position of the first stop is defined to be 0 and the buses reset their positions 0 when they reach the first stop. This section is based on previous work in [17].

1) Dynamics of Continuous States: The continuous states $x_i(t)$, $n_{i,j}(t)$, and $m_{h,j}(t)$ describe evolutions of bus positions, passenger accumulations inside buses, and those waiting at the stops.

(a) Bus position dynamics can be written as:

$$\begin{aligned} x_i(t+1) = \ & \textstyle\sum_{j=1}^{K_s} \gamma_{i,j}(t)\left(x_i(t) + Tv_i(t)\right) \\ & + \textstyle\sum_{j=2}^{K_s} \delta_{i,j}(t) x_i(t) + \delta_{i,1}(t)\cdot 0, \end{aligned} \tag{1}$$

for $i = 1, \ldots, K_b$, where t (–) is the time step counter, T (s) is the sampling time, $x_i(t)$ (m) and $v_i(t)$ (m/s) are the position and speed of bus i, respectively, $\gamma_{i,j}(t)$ is a binary state that is equal to 1 if bus i is cruising towards stop j and 0 otherwise, and $\delta_{i,j}(t)$ is a binary state that is equal to 1 if bus i is holding at stop j. The term $\sum_{j=1}^{K_s} \gamma_{i,j}(t)$ is 1 if bus i is cruising. If it is holding 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 remain the same. If it is holding at stop 1 (i.e., if $\delta_{i,1}(t) = 1$) this means that it has completed the loop and its position will be reset to 0.

(b) Bus accumulation dynamics can be written as:

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

for $i = 1, \ldots, K_b$ and $j = 1, \ldots, K_s$, where $n_{i,j}(t)$ (person) is the number of passengers in bus i destined to stop j, $q^{\text{in}}_{i,h,j}(t)$ (person/s) is the flow of passengers destined to stop j that are boarding bus i at stop h within a sampling period (i.e., in T seconds), whereas $q^{\text{out}}_{i,j}(t)$ (person/s) is the number of passengers alighting from bus i at stop j within a sampling period.

(c) Stop accumulation dynamics can be written as:

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

for $h, j = 1, \ldots, K_s$, $h \neq j$, where $m_{h,j}(t)$ (person) is the number of passengers at stop h destined to stop j, $\beta_{h,j}(t)$ (person/s) is the passenger flow arriving at stop h destined to stop j (i.e., the time varying origin-destination passenger flow demand from stop h to stop j). The accumulation at stop h increases with $T\beta_{h,j}(t)$ at each time step, whereas it may also decrease if there are bus(es) holding at stop h (i.e., if $\sum_{i=1}^{K_b} q^{\text{in}}_{i,h,j}(t)$ is non-zero) as passengers board the buses.

2) Dynamics of Binary States: To model parts of BTS dynamics that are impossible to describe with continuous variables, we introduce the binary states $\gamma_{i,j}(t)$ (*cruising state*) and $\delta_{i,j}(t)$ (*holding state*).

(a) Dynamics of the cruising state can be written as

$$\begin{aligned} \gamma_{i,j}(t+1) = \ & \big(\delta_{i,j-1}(t) \wedge \big(\big(e^m_{j-1}(t) \vee \\ & e^c_i(t)\big) \wedge e^n_{i,j-1}(t)\big)\big) \vee \big(\gamma_{i,j}(t) \wedge \neg e^x_{i,j}(t)\big), \end{aligned} \tag{4}$$

for $i = 1, \ldots, K_b$, $j = 1, \ldots, K_s$, where the event $e^m_j(t)$ is 1 if there are no passengers waiting at stop j and 0 otherwise, $e^c_i(t)$ is 1 if bus i is full and 0 otherwise, $e^n_{i,j}(t)$ is 1 if there are no passengers travelling on bus i destined to j and 0 otherwise, $e^x_{i,j}(t)$ is 1 if bus i has reached stop j and 0 otherwise. According to equation (4), bus i will start cruising to stop j if it is currently holding at stop $j-1$ ($\delta_{i,j-1}(t) = 1$), and there are no passengers on-board wanting to alight at stop $j-1$ ($e^n_{i,j-1}(t) = 1$), and either there are no passengers at stop $j-1$ boarding the bus ($e^m_j(t) = 1$) or the bus is full ($e^c_i(t) = 1$). The bus will continue to cruise if it is currently cruising ($\gamma_{i,j}(t) = 1$) and it has not yet reached stop j ($e^x_{i,j}(t) = 0$).

(b) Dynamics of holding state can be written as

$$\begin{aligned} \delta_{i,j}(t+1) = \ & \big(\gamma_{i,j}(t) \wedge e^x_{i,j}(t)\big) \vee \big(\delta_{i,j}(t) \wedge \\ & \neg\big(\big(e^m_j(t) \vee e^c_i(t)\big) \wedge e^n_{i,j}(t)\big)\big), \end{aligned} \tag{5}$$

for $i = 1, \ldots, K_b$, $j = 1, \ldots, K_s$. As per equation (5), bus i will start holding at stop j if it is currently cruising to stop j ($\gamma_{i,j}(t) = 1$) and it reaches stop j ($e^x_{i,j}(t) = 1$). It will continue to hold if it is currently holding ($\delta_{i,j}(t) = 1$), and there are passengers on-board wanting to alight ($e^n_{i,j}(t) = 0$), or there are passengers at the stop waiting to board ($e^m_j(t) = 0$) and the bus has available space ($e^c_i(t) = 0$).

3) Constraints Defining Events: (a) The event $e^x_{i,j}(t)$ describes the situation about whether bus i has passed stop j or not:

$$e^x_{i,j}(t) = \begin{cases} 0 & \text{if } x_i(t) \leq D_j \\ 1 & \text{otherwise} \end{cases} \tag{6}$$

for $i = 1, \ldots, K_b$, $j = 1, \ldots, K_s$, where D_j is the distance between stop j and stop 1 (with D_1 specifying the length of the whole loop).

(b) The event $e^m_h(t)$ expresses whether there any passengers at stop h or not:

$$e^m_h(t) = \begin{cases} 0 & \text{if } 0 < \sum_{j=1}^{K_s} m_{h,j}(t) \\ 1 & \text{otherwise} \end{cases} \quad \text{for } h = 1, \ldots, K_s. \tag{7}$$

(c) The event $e^c_i(t)$ keeps track of whether bus i is full or not:

$$e^c_i(t) = \begin{cases} 0 & \text{if } \sum_{j=1}^{K_s} n_{i,j}(t) < n_{i,\max} \\ 1 & \text{otherwise} \end{cases} \quad \text{for } i = 1, \ldots, K_b, \tag{8}$$

where $n_{i,\max}$ is the passenger capacity of bus i.

(d) The event $e^n_{i,j}(t)$ describes whether there are passengers on bus i that want to alight at stop j or not:

$$e^n_{i,j}(t) = \begin{cases} 0 & \text{if } 0 < n_{i,j}(t) \\ 1 & \text{otherwise,} \end{cases} \tag{9}$$

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

4) Constraints Defining Passenger Flows: (a) Passenger flow transferring from stop h to bus i cannot exceed the passenger transfer limit and may take positive values only when bus i is holding at stop h:

$$\textstyle\sum_{j=1}^{K_s} q^{\text{in}}_{i,h,j}(t) \leq \delta_{i,h}(t)\alpha T, \tag{10}$$

for $i = 1, \ldots, K_b$, $h = 1, \ldots, K_s$, where α (person/s) is a parameter (dictated by bus/stop geometry and size of the bus doors) expressing the maximum flow of passengers that may transfer between a bus and a stop.

(b) Passenger flow transferring from stop h to bus i cannot exceed the available space on bus i:

$$\textstyle\sum_{j=1}^{K_s} q^{\text{in}}_{i,h,j}(t) \leq n_{i,\max} - \sum_{j=1}^{K_s} n_{i,j}(t), \tag{11}$$

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

(c) Passenger flow transferring from stop h to the bus(es) holding at stop h destined to stop j cannot exceed the number of passengers currently at stop h destined to stop j:

$$\textstyle\sum_{i=1}^{K_b} q^{\text{in}}_{i,h,j}(t) \leq m_{h,j}(t), \tag{12}$$

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

(d) Passenger flow transferring from bus i to stop j cannot exceed the passenger transfer limit and may take positive values only when bus i is holding at stop j:

$$q^{\text{out}}_{i,j}(t) \leq \delta_{i,j}(t)\alpha T \tag{13}$$

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

(e) Passenger flow transferring from bus i to stop j cannot exceed the number of passengers on bus i wanting to alight at stop j:

$$q^{\text{out}}_{i,j}(t) \leq n_{i,j}(t) \tag{14}$$

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

B. Simulation of Bus Transport System

The MLD model describing BTS dynamics given in (1)-(5) and the events (6)-(9), enables simulations of a single loop BTS. The simulation requires bus speeds $v_i(t)$, passenger flow demands $\beta_{i,j}(t)$, and the initial states as input. Events are calculated by simply evaluating their definitions (e.g., at time t, if $n_{i,j}(t)$ has a non-zero value, then $e^n_{i,j}(t)$ is set to 0, otherwise it is set to 1). Passenger flows are determined via the following linear program:

$$\begin{array}{ll} \text{maximize} & \sum_{i=1}^{K_b}\sum_{j=1}^{K_s}\left(\sum_{h=1}^{K_s} q^{\text{in}}_{i,h,j}(t) + q^{\text{out}}_{i,j}(t)\right) \\ \text{subject to} & \text{constraints (10)–(14)} \\ & 0 \leq q^{\text{in}}_{i,h,j}(t), \\ & \quad \text{for } i = 1, \ldots, K_b,\ h, j = 1, \ldots, K_s,\ h \neq j \\ & 0 \leq q^{\text{out}}_{i,j}(t), \\ & \quad \text{for } i = 1, \ldots, K_b,\ j = 1, \ldots, K_s, \end{array} \tag{15}$$

where, at time t, the states $\delta_{i,h}(t)$, $n_{i,j}(t)$, and $m_{h,j}(t)$ are given constants. According to (15), passenger flows take their maximum possible values as restricted by bus capacity, maximum flow rates, number of passengers travelling in buses or waiting at stops, and, by the presence of buses holding at stops. The algorithm (1) formalizes the procedure for simulating the BTS.

Algorithm 1 Bus transport system simulation

1) At time step t, given the values of previous states $x_i(t)$, $n_{i,j}(t)$, $m_{h,j}(t)$, 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 $q^{\text{in}}_{i,h,j}(t)$ and $q^{\text{out}}_{i,j}(t)$ by solving the optimization problem (15) (or, via heuristics);
2) Given the values obtained at step 1, the previous states $x_i(t)$, $n_{i,j}(t)$, $m_{h,j}(t)$, $\delta_{i,j}(t)$, and $\gamma_{i,j}(t)$, and the exogenous inputs $v_i(t)$ and $\beta_{h,j}(t)$, evaluate the difference equations (1)-(5) to obtain the successor states $x_i(t+1)$, $n_{i,j}(t+1)$, $m_{h,j}(t+1)$, $\delta_{i,j}(t+1)$, and $\gamma_{i,j}(t+1)$.
3) Repeat steps 1 and 2 for $t = 1, \ \ldots, \ t_{\text{final}}$.

III. CONTROL DESIGN

Achieving schedule reliability requires keeping headways regular by minimizing their standard deviation, which is cumbersome to be directly integrated into a control framework. Regularizing bus spacings can be used as a proxy for achieving headway regularity, as an ideal BTS condition where speeds and spacings of all buses are equal corresponds to perfectly regular headways. In this section we propose a PI controller and a hybrid MPC scheme for regularizing bus spacings to achieve headway regularity.

A. PI Controller

For regular spacings, the ideal position of a bus is the middle point between the bus ahead of it and the bus behind it (i.e., the situation where the front and rear spacings are equal). The objective of keeping regular spacings can thus be formalized via the spacing error $e_i(t_c)$, defined as follows:

$$e_i(t_c) = x^{\text{a}}_i(t_c) + x^{\text{b}}_i(t_c) - 2x_i(t_c), \tag{16}$$

where $x^{\text{a}}_i(t_c)$ and $x^{\text{b}}_i(t_c)$ are the positions of buses ahead of and behind bus i, respectively, measured at control time step t_c. Based on the spacing error, a discrete-time PI controller can be formulated:

$$u_i(t_c) = u_i(t_c - 1) + K_{\text{P}} \cdot (e_i(t_c) - e_i(t_c - 1)) + K_{\text{I}} \cdot e_i(t_c), \tag{17}$$

where K_{P} and K_{I} are the gains. The PI controller is conceptually similar to the controller developed in [1] for $K_{\text{P}} = 0$.

B. Hybrid Model Predictive Controller

We propose here a novel hybrid MPC scheme capturing detailed dynamics of the BTS. The scheme is substantially different from the one proposed in [17]: The previous one included only a digital clock and bus speed constraints, whereas the scheme proposed here contains detailed hybrid dynamics including passenger accumulations in the bus and at the upcoming stop, the interaction between the bus and the stop via passenger flows, and the corresponding restriction on bus speeds (as the bus has to hold for passenger transfers).

Consider K_b buses operating on the loop, and $\tilde{K}_s$ active stops (a subset of all stops, totalling K_s) that are upcoming stops for the buses, where $\tilde{K}_s$ can at most be equal to K_b. This restriction on the number of considered stops is necessary to avoid excessive computational burden, as in BTSs the number of stops might be much larger than the number of buses, potentially leading to loss of real-time feasibility for the resulting hybrid MPC problems.

Bus position dynamics can be written as follows:

$$\begin{aligned} x_i(k+1) &= x_i(k) + T z_i(k) \\ x_{\mathrm{f},i}(k+1) &= x_{\mathrm{f},i}(k) + T(z_{i_\mathrm{f}}(k) - z_i(k)) \\ x_{\mathrm{r},i}(k+1) &= x_{\mathrm{r},i}(k) + T(z_i(k) - z_{i_\mathrm{r}}(k)), \end{aligned} \tag{18}$$

where k is the prediction time step counter, $x_i(k)$ is the position of bus i (defined to be zero at step $k = 1$), whereas $x_{\mathrm{f},i}(k)$ and $x_{\mathrm{r},i}(k)$ are the front and rear spacings of bus i (i.e., distances between bus i and the buses ahead of and behind bus i), respectively, with i_f and i_r being the indices of the buses ahead of and behind bus i, and $z_i(k)$ is the active speed of bus i, defined as follows:

$$z_i(k) = \begin{cases} v_i(k) & \text{if } \delta_{i,a}(k) = 0 \\ 0 & \text{otherwise,} \end{cases} \tag{19}$$

expressing the condition that bus i is restricted to have zero speed when it is holding (i.e., when $\delta_{i,a}(k) = 1$), where $\delta_{i,a}(k)$ is the holding state related to the first upcoming stop a, whereas $v_i(k)$ is the allowed speed for bus i, which is bounded by the minimum and maximum speeds:

$$v_{\text{min}} \le v_i(k) \le \bar{v}_i(k, t_c), \tag{20}$$

where the minimum bus speed v_{min} is a constant, and $\bar{v}_i(k, t_c)$ is the maximum speed of bus i that depends on the traffic conditions and is thus to be estimated at each control time step t_c. Allowed speed $v_i(k)$ also enables move blocking on the active bus speeds $z_i(k)$ as it is assumed that control sampling time T_c is an integer multiple of simulation sampling time T (which is also the prediction sampling time for the hybrid MPC), thus a corresponding move blocking is required.

Cruising state $\gamma_{i,a}(k)$ expresses whether bus i is cruising to first upcoming stop a or not, and evolves according to following dynamics:

$$\gamma_{i,a}(k+1) = \gamma_{i,a}(k) \wedge \neg e^x_{i,a}(k), \tag{21}$$

where the event $e^x_{i,a}(k)$ expresses whether bus i has reached stop a or not:

$$e^x_{i,a}(k) = \begin{cases} 1 & \text{if } d_{i,a}(t_c) - x_i(k) \le 0 \\ 0 & \text{otherwise,} \end{cases} \tag{22}$$

where $d_{i,a}(t_c)$ is the distance of bus i to stop a measured at time t_c.

Bus i will start holding once it reaches stop a, and the holding state dynamics can be written as:

$$\begin{aligned} \delta_{i,a}(k+1) = {} & \big(\gamma_{i,a}(k) \wedge e^x_{i,a}(k)\big) \vee \big(\delta_{i,a}(k) \\ & \wedge \neg\big((e^m_{i,a}(k) \vee e^c_i(k)) \wedge e^n_{i,a}(k)\big)\big), \end{aligned} \tag{23}$$

where the event $e^m_{i,a}(k)$ describes whether there are any passengers at stop a or not:

$$e^m_{i,a}(k) = \begin{cases} 1 & \text{if } m_{i,a}(k) \le 0 \\ 0 & \text{otherwise,} \end{cases} \tag{24}$$

with $m_{i,a}(k)$ being the accumulation state of passengers waiting at the first upcoming stop of bus i, whereas the event $e^c_i(k)$ expresses whether bus i has available space for passengers or not:

$$e^c_i(k) = \begin{cases} 1 & \text{if } n_{i,\text{max}} - n_i(k) \le 0 \\ 0 & \text{otherwise,} \end{cases} \tag{25}$$

with $n_i(k)$ being the accumulation state of passengers travelling on bus i, and the event $e^n_{i,a}(k)$ describes whether there are passengers on bus i destined to stop a or not:

$$e^n_{i,a}(k) = \begin{cases} 1 & \text{if } n_{i,a}(k) \le 0 \\ 0 & \text{otherwise,} \end{cases} \tag{26}$$

with $n_{i,a}(k)$ being the accumulation state of passengers travelling on bus i destined to stop a.

Dynamics of $m_{i,a}(k)$ can be written as follows:

$$m_{i,a}(k+1) = m_{i,a}(k) + \beta_a(k, t_c) - \sum_{i=1}^{\mathcal{I}^a} q^{\text{in}}_{i,a}(k), \tag{27}$$

where $\beta_a(k, t_c)$ is the passenger flow demands 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, and $q^{\text{in}}_{i,a}(k)$ is the boarding passenger flow transferring from stop a to bus i, defined through the following constraints:

$$\begin{aligned} 0 &\le q^{\text{in}}_{i,a}(k) \\ q^{\text{in}}_{i,a}(k) &\le \delta_{i,a}(k)\alpha T \\ q^{\text{in}}_{i,a}(k) &\le n_{i,\text{max}} - n_i(k). \end{aligned} \tag{28}$$

Dynamics of $n_i(k)$ and $n_{i,a}(k)$ are as follows:

$$\begin{aligned} n_i(k+1) &= n_i(k) + q^{\text{in}}_{i,a}(k) - q^{\text{out}}_{i,a}(k) \\ n_{i,a}(k+1) &= n_{i,a}(k) - q^{\text{out}}_{i,a}(k), \end{aligned} \tag{29}$$

where $q^{\text{out}}_{i,a}(k)$ is the alighting passenger flow transferring from bus i to stop a, defined through the following constraints:

$$\begin{aligned} 0 &\le q^{\text{out}}_{i,a}(k) \\ q^{\text{out}}_{i,a}(k) &\le \delta_{i,a}(k)\alpha T \\ q^{\text{out}}_{i,a}(k) &\le n_{i,a}(k). \end{aligned} \tag{30}$$

Initial bus position states can be written as follows

$$\begin{aligned} x_i(1) &= 0 \\ x_{\mathrm{f},i}(1) &= \tilde{x}_{i_\mathrm{f}}(t_c) - \tilde{x}_i(t_c) \\ x_{\mathrm{r},i}(1) &= \tilde{x}_i(t_c) - \tilde{x}_{i_\mathrm{r}}(t_c), \end{aligned} \tag{31}$$

where $x_i(1) = 0$ by definition and $\tilde{x}_i(t_c)$ is the measured position of bus i at control time step t_c, whereas initial cruising states are:

$$\gamma_{i,a}(1) = \tilde{\gamma}_{i,h_a(i)}(t_c), \tag{32}$$

where $\tilde{\gamma}_{i,h_a(i)}(t_c)$ is the measured cruising state of bus i at t_c with $h_a(i)$ being the index of the first upcoming stop of bus i, and initial holding states are

$$\delta_{i,a}(1) = \tilde{\delta}_{i,h_a(i)}(t_c), \tag{33}$$

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

$$m_{i,a}(1) = \sum_{j=1}^{K_s} \tilde{m}_{h_a(i),j}(t_c), \tag{34}$$

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

$$n_i(1) = \sum_{j=1}^{K_s} \tilde{n}_{i,j}(t_c) \tag{35}$$
$$n_{i,a}(1) = \tilde{n}_{i,h_a(i)}(t_c),$$

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

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:

$$\begin{aligned}
\underset{z_i(k)}{\text{minimize}} \quad & \sum_{i=1}^{K_b} \sum_{k=1}^{N} \left(y_{\mathrm{h},i}^2(k+1) + \sigma y_{\mathrm{z},i}^2(k) \right) \qquad (36)\\
\text{subject to} \quad & \text{for } i = 1, \ \ldots, \ K_b : \\
& \quad \text{initial states } (32), (33), (34), (35) \\
& \quad \text{for } k = 1, \ \ldots, \ N : \\
& \qquad \text{dynamics } (18), (21), (23), (27), (29) \\
& \qquad \text{constraints } (19), (20), (28), (30) \\
& \qquad \text{events } (22), (24), (25), (26)
\end{aligned}$$

where N is the prediction horizon, σ is the weight on fast operation term, whereas $y_{\mathrm{h},i}(k)$ and $y_{\mathrm{z},i}(k)$ are the spacing and speed errors for bus i, defined as follows:

$$y_{\mathrm{h},i}(k) = x_{\mathrm{f},i}(k) - x_{\mathrm{r},i}(k)$$
$$y_{\mathrm{z},i}(k) = \bar{v}_i(k, t_c) - z_i(k)$$

The problem (36) is a mixed-integer quadratic program (MIQP), which, although being a nonconvex problem, can be solved efficiently via software packages developed for mixed-integer programs.

The prediction model of the hybrid MPC assumes a certain scenario for the duration of the prediction horizon: In this scenario, bus i will start at the beginning (i.e., $k = 1$) either cruising to a or holding at a. As it is holding at a the passenger transfer procedures will take place, and once they are finished, bus i will once again allowed to cruise. Such a model expresses a version of the BTS simulation model given in section II that has a finite horizon in both time and space, with the finite time horizon 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 subsequent stops. Although an approximation of the full BTS dynamics, such an approach enables formulation of tractable hybrid MPC problems.

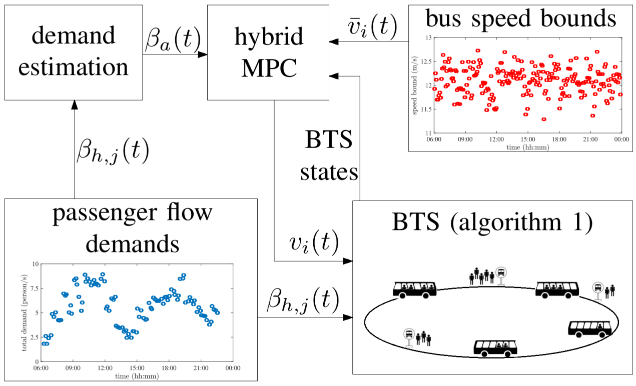

Fig. 1. Structure of the hybrid MPC scheme.

IV. SIMULATION RESULTS

A bus loop with 8 buses and 32 stops is considered, where each link is 1 km long. Simulation sampling time is $T = 10$ s and control sampling time is $T_c = 120$ s, whereas simulation length is chosen as $t_{\text{final}} = 6480$ steps, corresponding to 18 hours of BTS operation. The hybrid MPC scheme (depicted in Fig. 1) is compared with a no control (NC) case, where bus speeds are fixed to their maximum values $\bar{v}_i(t)$, and the PI controller, using the algorithm 1 for simulating BTS dynamics. Passenger demands are constructed with morning and evening peaks, and demands to certain stops are chosen higher than the rest for capturing spatial variability. Bus capacity is $n_{\max} = 80$ passengers, whereas the maximum passenger flow parameter is $\alpha = 0.5$ passenger/s. Absolute bounds on bus speeds are chosen as $v_{\min} = 4$ m/s and $v_{\max} = 20$ m/s, whereas the time-varying maximum bus speeds $\bar{v}_i(t)$ are generated randomly within the absolute bounds, to simulate the effect of traffic congestion. Prediction horizon of the hybrid MPC scheme is chosen as $N = 12$. Through trial-and-error, a weighting factor of $\sigma = 10800$ and PI controller gains of $K_{\mathrm{P}} = 1.5$ and $K_{\mathrm{I}} = 0.14$ are found to yield a good performance. The hybrid MPC scheme is implemented using the YALMIP toolbox [18], 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 [19] from YALMIP.

Bus positions on the loop as a function of simulation time and headway distributions, comparing the hybrid MPC scheme with the no control (NC) case and the PI controller, are shown in Fig. 2, whereas a summary of the

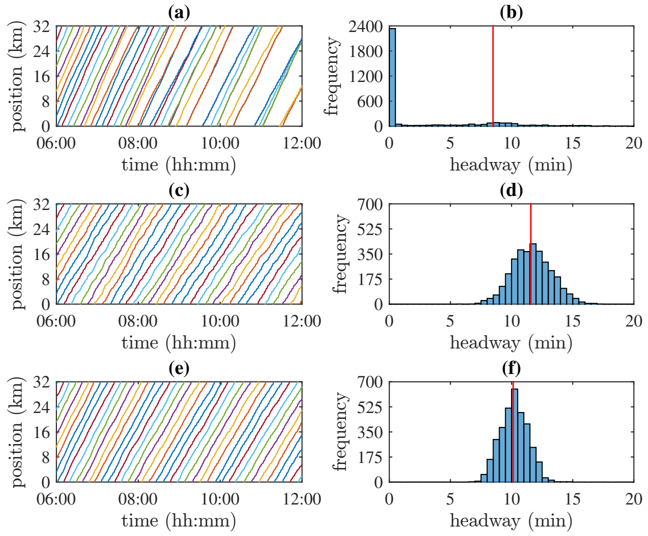

Fig. 2. Bus positions (a)-(c)-(e) as a function of time (showing only the first 6 hours of the scenario) and headway distributions (b)-(d)-(f) (vertical red lines show the mean values) for the no control case (a)-(b), the PI controller (c)-(d), and the hybrid MPC (e)-(f).

simulation results is given in table I, showing mean time spent per passenger (TSPP), mean commercial speed of buses, standard deviation of headways (std. of hws.; a metric of headway irregularity), and mean/maximum CPU times (only for the hybrid MPC). The results indicate that: (a) The PI yields reasonable performance and prevents bus bunching, (b) owing to a prediction model capturing detailed BTS dynamics, the proposed hybrid MPC shows superior performance compared to the PI, able to decrease mean TSPP while increasing headway regularity. The CPU times of the hybrid MPC suggest its real-time feasibility, as they are negligible in comparison to the control sampling time.

V. Conclusion

We designed a tractable hybrid MPC scheme with a detailed prediction model and tested its performance on simulation. Unlike the PI controller using only bus position measurements, the scheme exploits information on trip demands and traffic congestion using its prediction model considering the interactions between buses and stops via passenger flows. Yielding superior performance compared to the PI in both headway regularization and travel times, the hybrid MPC scheme shows potential for practice.

TABLE I
Control Performance Evaluation

Control scheme	mean TSPP (min)	mean c. speed (m/s)	std. of hws. (min)	mean/max. CPU time (s)
NC	34.5	7.7	–	–
PI	24.1	5.8	1.6	–
HMPC	20.7	6.6	1.1	0.6/3.5

Future work will include: (a) Comparison of the proposed scheme with other bus speed control methods from the literature, (b) more detailed simulation studies with various network configurations, (c) extension to multi-loop BTSs.

References

[1] C. F. Daganzo and J. Pilachowski, "Reducing bunching with bus-to-bus cooperation," *Transportation Research Part B: Methodological*, vol. 45, no. 1, pp. 267–277, 2011.

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

[3] M. D. Abkowitz and M. Lepofsky, "Implementing headway-based reliability control on transit routes," *Journal of Transportation Engineering*, vol. 116, no. 1, pp. 49–63, 1990.

[4] C. F. Daganzo, "A headway-based approach to eliminate bus bunching: Systematic analysis and comparisons," *Transportation Research Part B: Methodological*, vol. 43, no. 10, pp. 913–921, 2009.

[5] Y. Xuan, J. Argote, and C. F. Daganzo, "Dynamic bus holding strategies for schedule reliability: Optimal linear control and performance analysis," *Transportation Research Part B: Methodological*, vol. 45, no. 10, pp. 1831–1845, 2011.

[6] X. J. Eberlein, N. H. Wilson, and D. Bernstein, "The holding problem with real–time information available," *Transportation science*, vol. 35, no. 1, pp. 1–18, 2001.

[7] F. Delgado, J. C. Munoz, and R. Giesen, "How much can holding and/or limiting boarding improve transit performance?" *Transportation Research Part B: Methodological*, vol. 46, no. 9, pp. 1202–1217, 2012.

[8] R. Hall, M. Dessouky, and Q. Lu, "Optimal holding times at transfer stations," *Computers & industrial engineering*, vol. 40, no. 4, pp. 379–397, 2001.

[9] F. Delgado, N. Contreras, and J. C. Munoz, "Holding for transfers," in *Transportation Research Board 92nd Annual Meeting*, no. 13-2913, 2013.

[10] L. Fu, Q. Liu, and P. Calamai, "Real-time optimization model for dynamic scheduling of transit operations," *Transportation Research Record: Journal of the Transportation Research Board*, no. 1857, pp. 48–55, 2003.

[11] C. E. Cortés, D. Sáez, F. Milla, A. Núñez, and M. Riquelme, "Hybrid predictive control for real-time optimization of public transport systems' operations based on evolutionary multi-objective optimization," *Transportation Research Part C: Emerging Technologies*, vol. 18, no. 5, pp. 757–769, 2010.

[12] H. Liu, A. Skabardonis, and W.-b. Zhang, "A dynamic model for adaptive bus signal priority," in *Preprint CD-ROM, 82nd Transportation Research Board Annual Meeting, Washington, DC*, 2003.

[13] N. Van Oort, J. Boterman, and R. Van Nes, "The impact of scheduling on service reliability: trip-time determination and holding points in long-headway services," *Public Transport*, vol. 4, no. 1, pp. 39–56, 2012.

[14] P. Chandrasekar, R. Long Cheu, and H. C. Chin, "Simulation evaluation of route-based control of bus operations," *Journal of transportation engineering*, vol. 128, no. 6, pp. 519–527, 2002.

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

[16] A. Bemporad and M. Morari, "Control of systems integrating logic, dynamics, and constraints," *Automatica*, vol. 35, no. 3, pp. 407–427, 1999.

[17] I. Sirmatel and N. Geroliminis, "Dynamical modeling and predictive control of bus transport systems: A hybrid systems approach," vol. 50, no. 1, 2017, pp. 7499 – 7504, 20th IFAC World Congress. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S2405896317315537

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

[19] I. Gurobi Optimization, "Gurobi optimizer reference manual," 2016. [Online]. Available: http://www.gurobi.com