Technical report 09-039

# Hierarchical route choice control for baggage handling systems*

A.N. Tarău, B. De Schutter, and H. Hellendoorn

Delft Center for Systems and Control
Delft University of Technology
Mekelweg 2, 2628 CD Delft
The Netherlands
phone: +31-15-278.24.73 (secretary)
URL: https://www.dcsc.tudelft.nl

* This report can also be downloaded via https://pub.bartdeschutter.org/abs/09_039.html

# Hierarchical Route Choice Control for Baggage Handling Systems

Alina N. Tarău, Bart De Schutter, and Hans Hellendoorn

*Abstract*— **We propose a hierarchical control framework for state-of-the-art baggage handling systems where the luggage is transported by fast destination coded vehicles (DCVs). In this control framework switch controllers provide position instructions for each switch in the network. A collection of switch controllers is then supervised by a network controller that mainly takes care of the route choice instructions for DCVs. The route choice control problem is problem is a nonlinear, mixed integer optimization problem, with high computational requirements, which makes it intractable in practice. Therefore, we present an alternative approach for reducing the complexity of the computations by approximating the nonlinear optimization problem and rewriting it as a mixed integer linear programming (MILP) problem for which solvers are available that allow us to efficiently compute the global optimal solution. The solution of the MILP problem is then used in computing optimal switch control actions. For a benchmark case study we compare the hierarchical control with centralized switch control. The results indicate that the proposed hierarchical control offers a balanced trade-off between optimality and computational efficiency.**

## I. Introduction

State-of-the-art baggage handling systems in airports transport luggage at high speeds using destination coded vehicles (DCVs). These vehicles transport the bags in an automated way on a “mini” railway network. The first objective of a baggage handling system is to transport all the checked-in or transfer bags to the corresponding end points before the planes have to be loaded. However, due to the airport’s logistics, an end point is allocated to a plane only with a given amount of time before the plane’s departure. Hence, the baggage handling system performs optimally if each of the bags to be handled arrives at its given end point within a specific time window.

Currently, the DCVs are routed through the system using routing schemes based on preferred routes. These routing schemes can be adapted to respond on the occurrence of predefined events. However, as argued in [1], the patterns of loads on the system are highly variable, depending on e.g. the season, time of the day, type of aircraft at each gate, number of passengers for each flight. Therefore, in the research we conduct we do not consider predefined preferred routes. Instead we develop advanced control methods to determine the optimal routing in case of dynamic demand.

The route assignment problem has been addressed in e.g. [2], [3]. But, in our case we do not deal with a shortest-path or shortest-time problem, since we need the bags at their corresponding end point within a given time window. In [4] is presented an analogy between the DCV routing problems and data transmissions via internet. Also, [5] presents a multi-agent approach for routing DCVs. However, this multi-agent system is faced with major challenges due to the extensive communication required. The goal of our work is to develop and compare efficient control approaches for route choice control of each DCV on the track network.

Theoretically, the maximum performance of such a DCV-based baggage handling system would be obtained if one computes the optimal routes using optimal control [6]. However, as shown in [7], for a fast event-based model of this system, this control method becomes intractable in practice due to the heavy computation burden. Therefore, in order to make a trade-off between computational effort and optimality, in [8], we have also implemented centralized and decentralized model predictive control (MPC), and also a decentralized heuristic approach. As the results confirmed, centralized MPC requires high computation time to determine a solution. The use of decentralized control lowers the computation time, but at the cost of suboptimality.

We propose a hierarchical control framework where the higher level controllers use MPC. The large computation time obtained in previous work comes from solving the nonlinear, mixed integer optimization problems that have multiple local minima, and therefore, are difficult to solve. So, in this paper we investigate whether the computational effort required to compute the optimal route choice can be lowered by using mixed integer linear programming (MILP).

## II. Preliminaries

### A. DCV-based baggage handling systems

The track network of a DCV-based baggage handling systems consists of a set of loading stations as origin nodes, a set of unloading stations as destination nodes, and a set of junctions as internal nodes. Let us call the switch that makes the connection between a junction and its incoming links *switch-in*, and the switch that makes the connection between a junction and its outgoing links *switch-out*.

The DCV-based baggage handling system operates as follows: given a demand of bags and the network of tracks as a directed graph, the route of each DCV (from a given loading station to the corresponding unloading station) has to be determined subject to the operational and safety constraints detailed in [7] such that all the bags to be handled arrive at their end points within the corresponding time window.

All authors are with Faculty of Mechanical, Maritime and Materials Engineering, Delft Center for Systems and Control, Delft University of Technology, Mekelweg 2, 2628 CD Delft, The Netherlands a.n.tarau@tudelft.nl, b@deschutter.info, j.hellendoorn@tudelft.nl

Bart De Schutter is also with the Marine and Transport Technology department of Delft University of Technology.

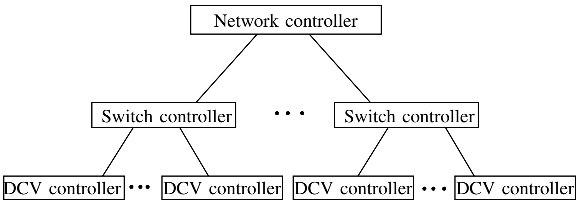

Fig. 1. Hierarchical control for DCV-based baggage handling systems.

### B. Control Framework

In order to efficiently compute the route choice of each DCV we propose a hierarchical control framework that consists of a multi-level control structure as shown in Fig. 1 with the following layers:

- The *network controller* provides the route choice for DCVs by determining reference flow trajectories over time for each link in the network. These flow trajectories are computed so that the performance of the system is optimized. Then the optimal reference flow trajectories are communicated to switch controllers.
- The *switch controller* present in each junction receives the information sent by the network controller and determines the sequence of optimal positions for its ingoing and outgoing switches at each time step so that the tracking error between the reference trajectory and the future flow trajectory is minimal.
- The *DCV controller* present in each vehicle detects the speed and position of the vehicle in front of it and the position of the switch into the junction the DCV travels towards to. This information is then used to determine the speed to be used next such that no collision will occur and such that the DCV stops in front of a junction the switch of which is not positioned on the link that the DCV travels.

The lower levels in this hierarchy deal with faster time scales (typically in the milliseconds range for the DCV controllers up to the seconds range for the switch controllers), whereas for the higher-level layer (network controller) the frequency of updating is up to the minutes range.

### C. Basic MPC

Since later on we will use the MPC approach for determining the routes of the DCVs in the network, in this section we briefly introduce the basic MPC concept.

MPC is an on-line model-based predictive control design method [9] that uses the receding horizon principle. In the basic MPC approach, given an horizon $N$, at step time $k$, the future control sequence $u(k+1), u(k+2), \ldots, u(k+N)$ is computed by solving a discrete-time optimization problem over a prediction period $[k\tau_{\mathrm{s}}, (k+N)\tau_{\mathrm{s}}]$ with $\tau_{\mathrm{s}}$ the sampling time and $k$ an integer. The optimization problem is defined so that a cost criterion is optimized over the prediction period subject to the operational constraints. After computing the optimal control sequence, only the first control sample is implemented, and subsequently the horizon is shifted. Next, the new state of the system is measured or estimated, and a new optimization problem at time $(k+1)\tau_{\mathrm{s}}$ is solved using this new information. In this way, a feedback mechanism is introduced.

## III. Route Choice Control

In this section we focus on the network controller. The switch controller will be discussed in Section IV.

### A. Approach

The predictive switch control problem results in a huge nonlinear integer optimization problem with high computational complexity and requirements, making the problem in fact intractable in practice [8]. So, since considering each individual switch is too computationally intensive we will consider streams of DCVs instead (characterized by real-valued demands and flows expressed in vehicles per second). The routing problem will be recast as the problem of determining the flows on each link. Once these flows are determined, they can be implemented by switch controllers at the junctions. So, the network controller provides flow targets to the switch controllers under its supervision, which then have to control the position of the switch into and out of each junction in such a way that these targets are met as well as possible.

### B. Set-up

We consider the following set-up. We have a transportation network with a set of origin nodes $\mathscr{O}$, a set of destination nodes $\mathscr{D}$, and a set of internal nodes $\mathscr{I}$. Define the set of all nodes as $\mathscr{V} = \mathscr{O} \cup \mathscr{I} \cup \mathscr{D}$. The nodes are connected by unidirectional links. Let $L$ denote the set of all links.

Let the time instant $t_k$ be defined as $t_k = k\tau^{\mathrm{nc}}$ with $\tau^{\mathrm{nc}}$ the sampling time for the network controller. Then, for each pair $(o,d) \in \mathscr{O} \times \mathscr{D}$, there is a dynamic, piecewise constant demand pattern $D_{o,d}(\cdot)$ with $D_{o,d}(k)$ the demand of bags at origin $o$ with destination $d$ in the time interval $[t_k, t_{k+1})$ for $k = 0, \ldots, K-1$ with $K$ the demand horizon (we assume that beyond $t_K$ the demand is 0). Let $L_d$ be the set of links that belong to some route going to $d$, $L_d \subseteq L$. We also denote the set of incoming links for node $v \in \mathscr{V}$ by $L_v^{\mathrm{in}}$, and the set of outgoing links by $L_v^{\mathrm{out}}$. Note that for origins $o \in \mathscr{O}$ we have $L_o^{\mathrm{in}} = \emptyset$ and for destinations $d \in \mathscr{D}$ we have $L_d^{\mathrm{out}} = \emptyset$. Also, without loss of generality, we assume each origin node to have only one outgoing link ($|L_o^{\mathrm{out}}| = 1$) and the destination nodes have only one incoming link ($|L_d^{\mathrm{in}}| = 1$) where $|\Lambda|$ represents the cardinality of the set $\Lambda$. For each destination $d \in \mathscr{D}$ and for each link $l \in L_d$ in the network we will define a real-valued flow $u_{l,d}(k)$. The flow $u_{l,d}(k)$ denotes the number of DCVs per time unit traveling towards destination $d$ that enter link $l$ during the time interval $[t_k, t_{k+1})$.

The aim is now to compute using MPC (see Section II), for each step $k$, flows $u_{l,d}(k)$ for every destination $d \in \mathscr{D}$ and for every link $l \in L_d$ in such a way that the capacity of the links is not exceeded and such that the performance criterion is minimized over a given prediction period $[t_k, t_{k+N}]$. Possible goals for the network controller that allow linear or piecewise affine performance criteria are reaching a desired outflow at destination $d$ or minimizing the lengths of the queue in the network.

### C. Route choice model

In this section we determine the model of the DCV flows through the network. Let $\tau_l$ denote the free-flow travel time on link $l$. The free-flow travel time represents the time period that a DCV requires to travel on a track segment in case of no congestion, using, hence, maximum speed. We assume the travel time $\tau_l$ to be an integer multiple of $\tau^{\mathrm{nc}}$.

In case the capacity of a loading station is less than the demand, queues might appear at the origin of the network. Let $q_{o,d}(k)$ denote the length of the partial queue of DCVs at origin $o$ going to destination $d$ at time instant $t_k$. In principle, the queue lengths should be integers as their unit is "number of vehicles", but we will approximate them using reals.

For every origin node $o \in \mathscr{O}$ and for every destination $d \in \mathscr{D}$ we now have:

$$u_{l,d}(k) \leqslant D_{o,d}(k) + \frac{q_{o,d}(k)}{\tau^{\mathrm{nc}}} \text{ for } l \in L_o^{\mathrm{out}} \cap L_d \qquad (1)$$

with $D_{o,d}(k) = 0$ for $k \geqslant K$. Moreover,

$$q_{o,d}(k+1) = \max(0, q_{o,d}(k) + (D_{o,d}(k) - \sum_{l \in L_o^{\mathrm{out}} \cap L_d} u_{l,d}(k))\tau^{\mathrm{nc}}) \quad (2)$$

But queues can form also inside the network. We assume that the DCVs run with maximum speed along the track segment and, if necessary, they wait before crossing the junction in a vertical queue. Let $q_{v,d}(k)$ denote the length of the vertical queue at junction $v \in \mathscr{I}$, for destination $d \in \mathscr{D}$, at time instant $t_k$. Note that, we do not consider outflow restrictions on queues to destination $d$ for a junction $v$ connected via a link to destination $d$ ($q_{v,d}(k) = 0$ for all $k$).

Taking into account that every flow on link $l$ has a delay of $\frac{\tau_l}{\tau^{\mathrm{nc}}}$ time steps before it reaches the end of the link, for every internal node $v \in \mathscr{I}$ and for every $d \in \mathscr{D}$ we have:

$$F_{v,d}^{\mathrm{out}}(k) \leqslant F_{v,d}^{\mathrm{in}}(k) + \frac{q_{v,d}(k)}{\tau^{\mathrm{nc}}} \qquad (3)$$

where $F_{v,d}^{\mathrm{in}}(k)$ is the flow into the queue at junction $v$, $F_{v,d}^{\mathrm{in}}(k) = \sum\limits_{l \in L_v^{\mathrm{in}} \cap L_d} u_{l,d}(k - \frac{\tau_l}{\tau^{\mathrm{nc}}})$ and where $F_{v,d}^{\mathrm{out}}(k)$ is the flow out of the queue at junction $v$, $F_{v,d}^{\mathrm{out}}(k) = \sum\limits_{l \in L_v^{\mathrm{out}} \cap L_d} u_{l,d}(k)$.

The evolution of the length of the queue for every internal node $v \in \mathscr{I}$ and for every $d \in \mathscr{D}$ is given by:

$$q_{v,d}(k+1) = \max(0, q_{v,d}(k) + (F_{v,d}^{\mathrm{in}}(k) - F_{v,d}^{\mathrm{out}}(k))\tau^{\mathrm{nc}}) \quad (4)$$

We also have the following condition for every link $l$:

$$\sum_{d \in \mathscr{D}} u_{l,d}(k) \leqslant U_l^{\max} \quad . \qquad (5)$$

where $U_l^{\max}$ is the maximum flow on link $l$.

Next we define the performance index to be used for computing the optimal routing at step $k$ for a prediction period $[t_k, t_{k+N})$.

The objective is to have each bag arriving at its end point within a given time interval $[t_d^{\mathrm{close}} - \tau_d^{\mathrm{open}}, t_d^{\mathrm{close}}]$ where $t_d^{\mathrm{close}}$ is the time instant when the end point $d$ closes and the last bags are loaded onto the plane, and $\tau_d^{\mathrm{open}}$ is the time period for which the end point $d$ stays open for a specific flight. We assume $t_d^{\mathrm{close}}$ and $\tau_d^{\mathrm{open}}$ to be integer multiples of $\tau_{\mathrm{s}}$. Without loss of generality, in this paper we consider that each destination has only one flight assigned to it. However, this can be easily extended to the general case, but where a presorting will be performed.

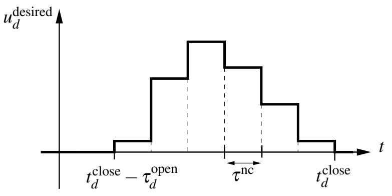

Fig. 2. Desired arrival profile at destination $d$.

Hence, one MPC objective that allows a piecewise affine performance criterion is to achieve a desired flow at destination $d$ during the prediction period. Let $u_d^{\mathrm{desired}}$ denote the desired piecewise constant flow profile at destination $d$ as sketched in Fig. 2, where the area under $u_d^{\mathrm{desired}}$ equals the total number of bags to be sent to destination $d$ out of the total demand. Note that outside the time window $[t_d^{\mathrm{close}} - \tau_d^{\mathrm{open}}, t_d^{\mathrm{close}})$ no bags should enter the incoming link of destination $d$ outside the given time window. Consequently, $u_d^{\mathrm{desired}}(k) = 0$ for all $k < k_d^{\mathrm{open}}$ and all $k > k_d^{\mathrm{close}}$ with $k_d^{\mathrm{open}} = \frac{t_d^{\mathrm{close}} - \tau^{\mathrm{open}}}{\tau_{\mathrm{s}}}$ and $k_d^{\mathrm{close}} = \frac{t_d^{\mathrm{close}}}{\tau_{\mathrm{s}}}$.

Hence, one can define the following penalty for flow profiles $J^{\mathrm{pen}}(k) = \sum\limits_{d \in \mathscr{D}} \lambda_d |u_d^{\mathrm{desired}}(k) - u_{l,d}(k + \frac{\tau_d^{\mathrm{dest}}}{\tau_{\mathrm{s}}})|$ where $\tau_d^{\mathrm{dest}}$ is the free-flow travel time of link $l \in L_d^{\mathrm{in}}$ and $\lambda_d > 0$ is a penalty that expresses the importance of the flight.

Note that using as MPC performance criterion $\sum_{i=k}^{k+N-1} J^{\mathrm{pen}}(i)$ for each time step $k$, could have adverse effects for small prediction horizons. Therefore, to counteract these effects, we also consider as additional controller goal maximizing the flows of all links that are not directly connected to unloading stations. To this aim, let $\tau_{l,d}^{\mathrm{link}}$ be the typical time required for a DCV that just entered link $l$ to reach destination $d$. Then one can define the following penalty: $J_{l,d}^{\mathrm{flow}}(k) = u_{l,d}(k)$ if $k_d^{\mathrm{open}} - \frac{\tau_{l,d}^{\mathrm{link}}}{\tau_{\mathrm{s}}} \leq k \leq k_d^{\mathrm{close}} - \frac{\tau_{l,d}^{\mathrm{link}}}{\tau_{\mathrm{s}}}$ and $J_{l,d}^{\mathrm{flow}}(k) = 0$ otherwise. This penalty will be later on used in the MPC performance criterion.

Next, in order to make sure that *all* the bags will be handled in finite time, we also include in the MPC performance criterion the weighted length of queues at each junction in the network as presented next. Let $\tau_{v,d}^{\mathrm{junc}}$ be the typical time required for a DCV in the queue at junction $v$ to reach destination $d$. Then we define the new penalty: $J_{v,d}^{\mathrm{overdue}}(k) = d_{v,d}^{\min} q_{v,d}(k)$ if $k \geq k_d^{\mathrm{close}} - \frac{\tau_{v,d}^{\mathrm{junc}}}{\tau_{\mathrm{s}}}$ and $J_{v,d}^{\mathrm{overdue}}(k) = 0$ otherwise, where $d_{v,d}^{\min}$ represents the length of the shortest route from junction $v$ to destination $d$.

Finally, let $L^{\mathrm{dest}}$ denote the set of links directly connected to unloading stations. Then the MPC performance index is defined as follows:

$$J_{k,N} = \sum_{i=k}^{k+N-1} \Big( J^{\text{pen}}(i) - \alpha \sum_{d\in\mathscr{D}} \sum_{l\in(L\setminus L^{\text{dest}})\cap L_d} J^{\text{flow}}_{l,d}(i) + \beta \sum_{d\in\mathscr{D}} \sum_{v\in\mathscr{I}} J^{\text{overdue}}_{v,d}(i) \Big)$$

with $\alpha \ll 1$ and $\beta \ll 1$ nonnegative weighting parameters.

Then the nonlinear MPC optimization problem is defined as follows:

$$\min_{\mathbf{u}(k)} J_{k,N} \qquad \text{s.t. (3)–(5).} \tag{6}$$

where $\mathbf{u}(k)$ is the control sequence consisting of all the flows $u_{l,d}(k) \ldots u_{l,d}(k+N-1)$ with $d \in \mathscr{D}$ and $l \in L_d$.

### *D. Equivalent MILP model*

In this section we transform the dynamic optimal route choice problem (6) into an MILP problem, for which efficient solvers have been developed [10]. To this aim we use the following equivalences, see [11], where $f$ is a function defined on a bounded set $X$ with upper and lower bounds $M$ and $m$ for the function values, $\delta$ is a binary variable, $y$ is a real-valued scalar variable, and $\varepsilon$ is a small tolerance (typically the machine precision):

**P1**: $[f(x) \leqslant 0] \iff [\delta = 1]$ is true if and only if

$$\begin{cases} f(x) \leqslant M(1-\delta) \\ f(x) \geqslant \varepsilon + (m-\varepsilon)\delta \end{cases},$$

**P2**: $y = \delta f(x)$ is equivalent to

$$\begin{cases} y \leqslant M\delta \\ y \geqslant m\delta \\ y \leqslant f(x) - m(1-\delta) \\ y \geqslant f(x) - M(1-\delta) \end{cases}.$$

As an example we will show how equation (2) of the nonlinear route choice model presented in the previous section can be transformed into a set of linear equations and inequalities by introducing some auxiliary variables. For the other equations of the route choice model we apply a similar procedure.

Depending on the order in which properties **P1** and **P2** are applied and in which additional auxiliary variables are introduced, we may end up with more or less binary and real-valued variables in the final MILP problem. The number of binary variables — and to a lesser extent the number of real variables — should be kept as small as possible since this number has a direct impact on the computational complexity of the final MILP problem.

We consider now (2). This is a nonlinear equation and thus it does not fit the MILP framework. Therefore, we will first introduce the binary variables $\delta_{o,d}(k)$ such that

$$\delta_{o,d}(k) = 1 \text{ if and only if}$$
$$q_{o,d}(k) + \big(D_{o,d}(k) - \sum_{l\in L_o^{\text{out}}\cap L_d} u_{l,d}(k)\big)\tau^{\text{nc}} \leqslant 0 \tag{7}$$

and rewrite (2) as follows:

$$q_{o,d}(k+1) = \big(1-\delta_{o,d}(k)\big)\cdot\Big(q_{o,d}(k) + \big(D_{o,d}(k) - \sum_{l\in L_o^{\text{out}}\cap L_d} u_{l,d}(k)\big)\tau^{\text{nc}}\Big). \tag{8}$$

Condition (7) is equivalent to (cf. Property **P1**):

$$\begin{cases} f(k) \leqslant (q^{\max}_{o,d} + D^{\max}_{o,d}\tau^{\text{nc}})(1-\delta_{o,d}) \\ f(k) \geqslant \varepsilon + (-U^{\max}_{l}\tau^{\text{nc}} - \varepsilon)\delta_{o,d} \end{cases},$$

where $f(k) = q_{o,d}(k) + \big(D_{o,d}(k) - u_{l,d}(k)\big)\tau^{\text{nc}}$ with $l \in L_o^{\text{out}} \cap L_d$, $U_l^{\max}$ is the maximal possible flow out of origin node $o$ towards destination $d$, $q^{\max}_{o,d}$ is the maximal queue length at origin $o$ for traffic going to destination $d$, and where $D^{\max}_{o,d} = \max_k D_{o,d}(k)$ is the maximal demand for origin-destination pair $(o,d)$.

However, (8) is still nonlinear since it contains a multiplication of a binary variable $\delta_{o,d}(k)$ with a real-valued (linear) function. However, by using Property **P2** this equation can be transformed into a system of linear inequalities.

Next we transform the nonlinear terms of (6) into sets of equality and inequality constraints. For example the problem

$$\min_{\mathbf{u}(k)} \sum_{d\in\mathscr{D}} \lambda_d \sum_{i=k}^{k+N-1} \Big| u_d^{\text{desired}}(k) - \sum_{l\in L_d^{\text{in}}} u_{l,d}\big(k + \tfrac{\tau_d^{\text{dest}}}{\tau_{\text{s}}}\big)\Big|$$

can be written as:

$$\begin{aligned} &\min \sum_{d\in\mathscr{D}} \sum_{i=k}^{k+N-1} \lambda_d u_d^{\text{diff}}(i) \\ &\text{s.t.} \\ &u_d^{\text{diff}}(i) \geqslant u_d^{\text{desired}}(i) - \sum_{l\in L_d^{\text{in}}} u_{l,d}\big(i + \tfrac{\tau_d^{\text{dest}}}{\tau_{\text{s}}}\big) \\ &\qquad \text{for } i = k+1,\ldots,k+N-1 \\ &u_d^{\text{diff}}(i) \geqslant -u_d^{\text{desired}}(i) + \sum_{l\in L_d^{\text{in}}} u_{l,d}\big(i + \tfrac{\tau_d^{\text{dest}}}{\tau_{\text{s}}}\big) \\ &\qquad \text{for } i = k+1,\ldots,k+N-1, \end{aligned}$$

which is a linear programming problem.

So, the overall objective function $J_{k,N}$ can be written as a linear one. Hence, the problem (6) can be written as an MILP problem.

Several efficient branch-and-bound MILP solvers [10] are available for MILP problems. In principle, — i.e., when the algorithm is not terminated prematurely due to time or memory limitations, — these algorithms guarantee to find the global optimum. This global optimization feature is not present in the other optimization methods that can be used to solve the original nonlinear, nonconvex, nonsmooth optimization problem (6). Moreover, if the computation time is limited (as is often the case in on-line real-time control), then it might occur that the MILP solution can be found within the allotted time whereas the global and multi-start local optimization algorithm still did not converge to a good solution. As a result, the MILP solution even performs much better than the solution returned by the prematurely terminated global and multi-start local optimization method.

Hence, we can say that the MILP solution often provides a good trade-off between optimality and computational effort, as will be illustrated in the case study of Section V.

# IV. Switch Control

In this section we focus on the switch controller for the proposed hierarchy.

Recall that at each control step $k$, the network controller provides optimal flows for each link in the network and for each destination. Let these flows be denoted by $u_{l,d}^{\text{opt}}(k)$ with $d \in \mathscr{D}$ and $l \in L \cap L_d$. Then the switch controller of each junction has to compute optimal switch-in and switch-out positions such that the tracking error between the reference optimal flow trajectory and the flow trajectory obtained by the switch controller is minimal for each network controller time step $k = 0, \ldots, K_{\text{sim}}$. Next we will refer to one junction $v \in \mathscr{I}$ only. For all other junctions, the switch control actions are determined similarly.

Let $s_v^{\text{in}}(k^{\text{sc}})$ denote the position of the switch-in at junction $v \in \mathscr{I}$ during the time interval $\left[t_{k^{\text{sc}}}^{\text{sw}}, t_{k^{\text{sc}}+1}^{\text{sw}}\right)$, where $t_{k^{\text{sc}}}^{\text{sw}} = t_k + k^{\text{sc}}\tau^{\text{sc}}$ with $k^{\text{sc}}$ an integer and $\tau^{\text{sc}}$ the switch controller sampling time ($t_k = t_0^{\text{sc}}$). Similarly, we define $s_v^{\text{out}}(k^{\text{sc}})$, the position of the switch-out at junction $v \in \mathscr{I}$ during the time interval $\left[t_{k^{\text{sc}}}^{\text{sw}}, t_{k^{\text{sc}}+1}^{\text{sw}}\right)$.

We want to determine the switch control sequence at most until time instant $t_{k+1}$. However, the prediction period has at most $N_{\max}^{\text{sc}}$ steps. As a consequence, the prediction period for the MPC switch problem at step $k^{\text{sc}}$ is defined as $\left[t_{k^{\text{sc}}}^{\text{sw}}, t_{\text{end},k}^{\text{sw}}\right)$ with $t_{\text{end},k}^{\text{sw}} = \min(t_{k+1}, t_{k^{\text{sc}}+N^{\text{sc}}}^{\text{sw}})$.

Hence, at each MPC step $k^{\text{sc}}$, the switch controller solves the optimization problem: $\min_{\mathbf{s}_v} \; J_{k^{\text{sc}},N^{\text{sc}}}^{\text{sw},v,k}(\mathbf{s}_v)$ where

- $N^{\text{sc}}$ is the length of the prediction horizon ($N^{\text{sc}} = \frac{t_{\text{end},k}^{\text{sw}}}{\tau^{\text{sc}}}$),
- $\mathbf{s}_v = [s_v^{\text{in}}(k^{\text{sc}}) \ldots s_v^{\text{in}}(k^{\text{sc}} + N^{\text{sc}} - 1) \ldots s_v^{\text{out}}(k^{\text{sc}}) \ldots s_v^{\text{out}}(k^{\text{sc}} + N^{\text{sc}} - 1)]^{\text{T}}$,
- $J_{k^{\text{sc}},N^{\text{sc}}}^{\text{sw},v,k}$ is the local performance index defined next.

Let $X_{v,l,k}^{\text{opt}}$ denote the optimal number of DCVs to enter the outgoing link $l$ of junction $v$ during the period $\left[t_{k^{\text{sc}}}^{\text{sw}}, t_{\text{end},k}^{\text{sw}}\right)$. Then for $l \in L_v^{\text{out}} \cap L_d$ the variable $X_{v,l,k}^{\text{opt}}$ is given by $X_{v,l,k}^{\text{opt}} = (t_{\text{end},k}^{\text{sw}} - t_{k^{\text{sc}}}^{\text{sw}}) \sum_{d \in \mathscr{D}} u_{l,d}^{\text{opt}}(k)$. Next let $X_{v,l,k^{\text{sc}}}$ be the actual number of DCVs entering link $l$ during the prediction period. The variable $X_{v,l,k^{\text{sc}}}$ is determined via simulation for the fast event-driven model of [8]. Then, at time step $k^{\text{sc}}$, the local performance index is defined as follows:

$$J_{k^{\text{sc}},N^{\text{sc}}}^{\text{sw},v,k}(\mathbf{s}_v) = \sum_{l \in L_v^{\text{out}} \cap L_d} |X_{v,l,k}^{\text{opt}} - X_{v,l,k^{\text{sc}}}(\mathbf{s}_v)| + \gamma\big(n^{\text{sw\_in}}(\mathbf{s}_v) + n^{\text{sw\_out}}(\mathbf{s}_v)\big)$$

where $n^{\text{sw\_in}}$ and $n^{\text{sw\_out}}$ represent the number of toggles of the switch-in and of the switch-out respectively during the prediction period $\left[t_{k^{\text{sc}}}^{\text{sw}}, t_{\text{end},k}^{\text{sw}}\right)$, which are obtained from simulation, and where $\gamma$ is a nonnegative weighting parameter.

# V. Case Study

## A. Set-up and scenarios

We consider the network of tracks depicted in Fig. 3 with 4 loading stations, 2 unloading stations, 9 junctions, and 20 unidirectional links, where the free-flow travel time is provided for each link. This network allows more than four possible routes to each destination from any origin point. We consider this network because on the one hand it is simple, allowing an intuitive understanding of and insight in the operation of the system and the results of the control, and because on the other hand, it also contains all the relevant elements of a real set-up.

We assume that the velocity of each DCV varies between 0 m/s and 10 m/s. In order to faster assess the efficiency of our control method we assume that we do not start with an empty network but with a network already populated by DCVs transporting bags.

We consider 6 different scenarios where 2500 bags have to be handled for different initial states of the system, queues on different links, different piecewise constant demand profiles over the first 180 s of the simulation, and different weighting parameters. We simulate a period of 40 min. The control time step for the network controller is set to 60 s, while the control time step for the switch controller is set to 2 s. In these scenarios we have also considered the occurrence of queues at origin. Assuming that we start the simulation at time instant $t_0 = 0$ s, we consider the time window to be $[800, 1400]$ for destination $d_1$, and $[1000, 1600]$ for destination $d_2$.

## B. Results

In this section we compare the results obtained when using the proposed hierarchical control framework and the centralized switch control of [8]. In order to solve the MILP optimization of the network controller we have used the CPLEX solver of the Matlab optimization toolbox *Tomlab*, while to solve the nonlinear optimization problem of the switch controller we have chosen a *genetic* algorithm of the Matlab optimization toolbox *Genetic Algorithm and Direct Search* implemented via the function `ga`, using multiple runs. The same genetic algorithm has been used to solve the optimization problem of the centralized switch control. Essentially the centralized switch control boils down to solving a problem like (6) but with $r(k+1), r(k+2), \ldots, r(k+N)$ as optimization variables for each MPC step $k$, where $r(i)$ is the route of the $i$th DCV that entered the network (for details see [8]). As prediction horizon we have considered $N = 11$ for the network controller and $N^{\text{sc}} = 15$ for the switch controller of the hierarchical control, and $N = 40$ for the centralized MPC switch control. Note that due to

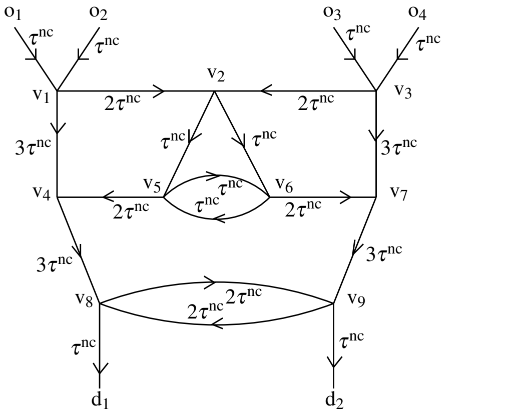

Fig. 3. Case study for a DCV-based baggage handling system.

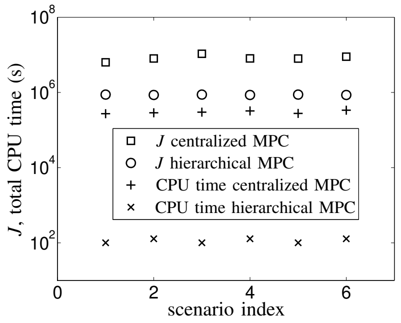

Fig. 4. Closed-loop results (the smaller $J$ the better system performance).

computational requirements reasons, for the switch control of both frameworks we shift the horizon with $N$, respectively $N^{\text{sc}}$ samples at each MPC step. Also, due to the same reason (computational requirements), we allow a limited amount of time (1 hour) for solving an optimization problem corresponding to the centralized switch control.

Based on simulations we now compare, for the given scenarios, the results obtained for the proposed control frameworks. The results of the simulations are reported in Fig. 4. For this comparison we consider the total performance of the system to be defined as $J = \sum_{d \in \mathscr{D}} \sum_{i=1}^{X_d} |t_{i,d} - t_{i,d}^{\text{desired}}|$ with $t_{i,d}$ the time when the $i$th bag crossing the junction directly connected to destination $d$ actually crosses that junction, $t_{i,d}^{\text{desired}}$ is the desired crossing time for the same DCV, and $X_d$ the total number of bags to be sent to destination $d$ during the simulation period. The time sequence $t_{1,d}^{\text{desired}}, \ldots, t_{X_d,d}^{\text{desired}}$ with $d \in \mathscr{D}$ is computed such that at each control time step $k$ of the network controller, the $\tau^{\text{nc}} u_d^{\text{desired}}(k)$ bags arrive at equidistant time instants during the period $[t_k, t_{k+1})$.

Using simulations we have obtained an average performance over all scenarios of $9.92 \cdot 10^5$ s for the hierarchical control framework versus a performance of $6.44 \cdot 10^6$ s. So, simulation results confirm that computing the route choice using the hierarchical control framework gives better performance than using the centralized switch control. Hence, the hierarchical control with MILP flow solutions performs better than the centralized switch control, the solution of which was returned by the prematurely terminated global and multi-start local optimization method.

However, even with these computational restrictions, the total computation time of the centralized switch control (over 62 hours) is much larger than the one of the hierarchical control (an average of 246 s per junction, plus 12 s for solving the MILP optimization problems).

Hence, the proposed hierarchical control outperforms the centralized switch control of [8].

## VI. Conclusions

In this paper we have proposed a hierarchical control framework for efficiently computing routes for destination coded vehicles (DCVs) that transport bags in an airport on a railway network. In the proposed control framework the network controller computes reference flow trajectories over time for each link in the network so that the performance of the DCV-based baggage handling system is optimized. Then the switch controllers determine the sequence of optimal positions for their ingoing and outgoing switches so that the tracking error between the reference trajectory and the future flow trajectory is minimal. The problem of computing optimal routes for DCVs is a nonlinear, nonconvex, mixed integer optimization problem, and very expensive to solve in terms of computational efforts. Therefore, we have used an alternative approach for reducing the complexity of the computations by rewriting the nonlinear optimization problem of the network controller as a mixed integer linear programming (MILP) problem. The advantage is that for MILP optimization problems solvers are available that allow us to efficiently compute the global optimal solution. The solution of the MILP problem is then used in computing optimal switch control actions. For a benchmark case study we have compared the hierarchical control with centralized switch control. Results indicate that the proposed hierarchical control outperforms the centralized switch control where the multi-start local optimization method has been terminated prematurely.

## Acknowledgment

This research is supported by the STW-VIDI project "Multi-Agent Control of Large-Scale Hybrid Systems", by BSIK project "Next Generation Infrastructures", Transport Research Centre Delft, Delft Research Centre Next Generation Infrastructures, and by the European STREP project "Hierarchical and Distributed Model Predictive Control".

## References

[1] R. de Neufville, "The baggage system at Denver: Prospects and lessons," *Journal of Air Transport Management*, vol. 1, no. 4, pp. 229–236, Dec. 1994.

[2] H. Gang, J. Shang, and L. Vargas, "A neural network model for the free-ranging AGV route-planning problem," *Journal of Intelligent Manufacturing*, vol. 7, no. 3, pp. 217–227, 1996.

[3] D. Kaufman, J. Nonis, and R. Smith, "A mixed integer linear programming model for dynamic route guidance," *Transportation Research Part B: Methodological*, vol. 32, no. 6, pp. 431–440, 1998.

[4] A. Fay, "Decentralized control strategies for transportation systems," in *Proceedings of the 2005 IEEE International Conference on Control and Automation*, Budapest, Hungary, Jun. 2005, pp. 898–903.

[5] K. Hallenborg and Y. Demazeau, "Dynamical control in large-scale material handling systems through agent technology," in *Proceedings of the 2006 IEEE /WIC/ACM International Conference on Intelligent Agent Technology*, Hong Kong, China, Dec. 2006, pp. 637–645.

[6] F. Lewis, *Optimal Control.* New York, New York, USA: John Wiley & Sons, Inc., 1986.

[7] A. Tarău, B. De Schutter, and J. Hellendoorn, "Travel time control of destination coded vehicles in baggage handling systems," in *Proceedings of the 17th IEEE International Conference on Control Applications*, San Antonio, Texas, USA, Sep. 2008, pp. 293–298.

[8] ——, "Route choice control of automated baggage handling systems," in *Proceedings of the 88th Annual Meeting of the Transportation Research Board*, Washington DC, USA, Jan. 2009, paper 09-0432.

[9] E. Camacho and C. Bordons, *Model Predictive Control in the Process Industry.* Berlin, Germany: Springer-Verlag, 1995.

[10] R. Fletcher and S. Leyffer, "Numerical experience with lower bounds for MIQP branch-and-bound," *SIAM Journal on Optimization*, vol. 8, no. 2, pp. 604–616, May 1998.

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