Delft University of Technology

Delft Center for Systems and Control

Technical report 09-031

Decentralized route choice control of automated baggage handling systems*

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

If you want to cite this report, please use the following reference instead:
A.N. Tarău, B. De Schutter, and J. Hellendoorn, "Decentralized route choice control of automated baggage handling systems," *Proceedings of the 12th IFAC Symposium on Transportation Systems*, Redondo Beach, California, pp. 70–75, Sept. 2009.

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.deschutter.info/abs/09_031.html

Decentralized route choice control of automated baggage handling systems

A.N. Tarău * **B. De Schutter** *,** **J. Hellendoorn** *

* *Delft Center for Systems and Control*
Delft University of Technology, Mekelweg 2, 2628 CD Delft, The Netherlands
{*a.n.tarau, j.hellendoorn*}*@tudelft.nl*
** *Marine and Transport Technology Department*
Delft University of Technology, The Netherlands, b@deschutter.info

Abstract: Modern baggage handling processes in airports use destination coded vehicles (DCVs) to transport the luggage at high speeds. These vehicles transport the bags on a "mini" railway network. In order optimize the performance of a DCV-based baggage handling system, the route of each DCV has to be determined. In this paper we consider an event-based model of this system. For routing the DCVs through the network we propose decentralized control methods that independently compute local control actions viz. decentralized model predictive control (MPC) and decentralized heuristic approaches. The considered control methods are compared for several scenarios. Results indicate that decentralized MPC can be used to suboptimally solve the problem. Moreover, the decentralized heuristic approaches usually give worse results than those obtained when using decentralized MPC, but on the other hand they require very low computation time.

Keywords: Baggage handling systems, decentralized route choice control, discrete-event simulation.

1. INTRODUCTION

The increasing need for reductions of costs of the air transport industry and the rise of low-cost carriers require a cost effective operation of the airports. The state-of-the-art technology used in a baggage handling system to transport the bags in an automated way incorporates scanners that scan the labels on each piece of luggage, baggage screening equipment for security scanning, networks of conveyors equipped with junctions that route the bags through the system, and destination coded vehicles (DCVs). As illustrated in Figure 1, a DCV is a metal cart with a plastic tub on top. These carts transport the bags at high speed on a "mini" railway network.

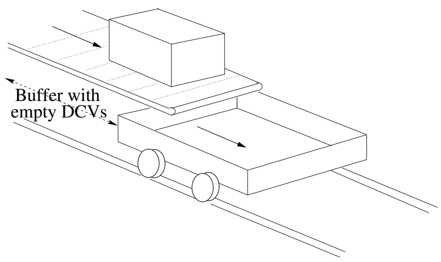

Fig. 1. Loading a DCV.

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.

In this paper we consider a DCV-based baggage handling system. Higher-level control problems are route assignment for each bag (and implicitly the switch control of each junction), line balancing (route assignment for each empty DCV such that all the loading stations have enough empty DCVs at any time instant), and prevention of buffer overflows. The low-level control problems of this system are velocity control of each DCV, coordination and synchronization when loading a bag onto a DCV, in order to avoid damaging the bags or blocking the system, and when unloading it to the corresponding end point. Controllers that solve these low-level problems are assumed to be present in the system.

We consider higher-level control problems. In particular, in this paper we focus on the route choice of DCVs that transport the bags. In the literature, the route assignment problem has been addressed by e.g. Gang et al. (1996), Kaufman et al. (1998). 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. Centralized control methods for a DCV-based baggage handling system require large computational effort see e.g. (Tarău et al., 2008). Therefore, the goal of this paper is to develop and compare efficient decentralized approaches for controlling the route choice of the DCVs.

The paper is organized as follows. In Section 2, an event-driven model of the system is briefly presented. Afterwards, in Section 3 and Section 4, we propose several control methods to determine the route of each DCV in a decentralized manner. The proposed control methods are: decentralized model predictive control and two decentralized heuristic approaches. The analysis of the simulation results and the comparison of the proposed control methods are elaborated in Section 5. Finally, the conclusions for this paper are drawn in Section 6.

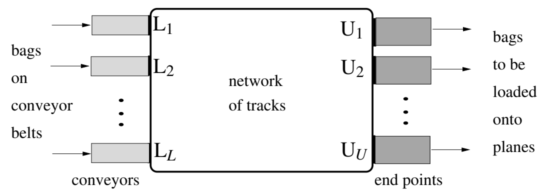

Fig. 2. Baggage handling system using DCVs.

2. SYSTEM DESCRIPTION AND MODEL

We use the general DCV-based baggage handling system sketched in Figure 2. This system operates as follows: given a demand of bags (identified by their unique code) together with their arrival times at the loading stations, and the network of tracks, the route of each DCV (from a given loading station to the corresponding unloading station) has to be computed such that the performance of the system is optimized subject to operational and safety constraints. We assume that the velocity of each DCV is always at its maximum unless overruled by the local on-board collision avoidance controller. These on-board collision avoidance controllers ensure a minimum safe distance between DCVs and also hold DCVs at switching points, if required.

We assume that there is a sufficient number of DCVs present in the system so that when a bag is at the loading station there is always a DCV ready to transport it. Also, we consider that the capacity of the end points is large enough that no buffer overflow can occur.

In (Tarău et al., 2008) we have developed an event driven model of the DCV-based baggage handling system. This model consists of a continuous part describing the movement of the individual vehicles transporting the bags through the network, and of the following discrete events: loading a new bag onto a DCV, unloading a bag that arrives at its end point, updating the position of the switch going into a junction (called switch-in hereafter) and the position of a switch going out of a junction (called switch-out hereafter), and updating the speed of a DCV. For details we refer to (Tarău et al., 2008).

The operational constraints are derived from the mechanical and design limitations e.g. the position of a switch at a junction can only change after minimum τ_{x} time units in order to avoid the quick and repeated back and forth movements of the switch, which may lead to mechanical damage.

As shown by Tarău et al. (2008), centralized approaches to determine the optimal route of the DCVs become intractable in practice due to the large computational effort required. Therefore, in order to lower the computational effort, in the sequel we propose decentralized approaches to control the route of the DCVs such as decentralized model predictive control and fast heuristic approaches.

3. DECENTRALIZED MODEL PREDICTIVE CONTROL

Model predictive control (MPC) is an on-line model-based control design method that uses the receding horizon principle Maciejowski (2002). This control approach can be used for DCV-based baggage handling systems, see.(Tarău et al., 2008). In this section we define the local MPC problem.

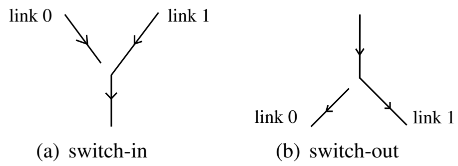

Fig. 3. Incoming and outgoing links at a junction. The switch-in and switch-out are positioned on link 1.

3.1 Local system boundaries

In decentralized MPC we consider local systems, each consisting of a junction S_s with $s \in \{1,2,\ldots,S\}$, its incoming and its outgoing links. Note that without loss of generality we can assume that each junction has maximum 2 incoming links and maximum 2 outgoing links, both indexed by $l \in \{0,1\}$ as sketched in Figure 3. For the sake of simplicity of notation, in the remainder of this section, we will not explicitly indicate the subscript s for variables that refer to junction S_s since we refer to one junction only. For all the other junctions, the same procedure is applied.

3.2 Local control measures

In contrast to basic MPC, we do not use a time index, but a bag index. So, we index the bags that successively cross a junction S_s during the entire simulation period as $b_1, b_2, \ldots, b_{N_{\mathrm{bags}}}$, where N_{bags} is the number of bags that cross S_s during the simulation period.

In this section we control the positions of the switch-in and switch-out of junction S_s for each bag that crosses S_s. The local control will be updated every time some bag has just crossed a junction. Let t_{crt} denote a time instant at which the local controls are updated. For junction S_s we now determine bag step k such that $t_{\mathrm{cross},k} \leq t_{\mathrm{crt}} < t_{\mathrm{cross},k+1}$, where $t_{\mathrm{cross},k}$ is defined as the time instant when bag b_k has just crossed the junction.

The local optimization is performed over the next $N \leq N_{\mathrm{bags}}$ bags that pass junction S_s after bag step k. By solving this local optimization problem we compute the control sequence $\mathbf{u}(k) = [u_{\mathrm{sw\_in}}(k+1) \ldots u_{\mathrm{sw\_in}}(k+N)\, u_{\mathrm{sw\_out}}(k+1) \ldots u_{\mathrm{sw\_out}}(k+N)]^{\mathrm{T}}$ corresponding to the next N bags $b_{k+1}, b_{k+2}, \ldots, b_{k+N}$ that will cross the junction. The control decisions $u_{\mathrm{sw\_in}}(k+1), \ldots, u_{\mathrm{sw\_in}}(k+N)$ of the switch into S_s determine the order in which the bags cross the junction and the corresponding time instants at which the bags $b_{k+1}, \ldots, b_{k+N}$ enter S_s. The control decisions $u_{\mathrm{sw\_out}}(k+1), \ldots, u_{\mathrm{sw\_out}}(k+N)$ determine the next junction towards which the bags $b_{k+1}, \ldots, b_{k+N}$ will travel.

3.3 Local objective function

When solving the local MPC optimization problem, we will use a local objective function $J_{\mathrm{DMPC},N}$. The local objective function is computed via a simulation of the local system for the next N bags that will cross the junction. This computation is performed as follows. We first define the performance index $J_{\mathrm{pen},b_{k+j}}$ for bag b_{k+j}, $j = 1,2,\ldots,N$. This performance index penalizes the overdue time and the additional storage time:

$$\begin{aligned}
&J_{\mathrm{pen},b_{k+j}}(t_{\mathrm{unload},b_{k+j}}) = \\
&\quad \sigma_{b_{k+j}} \max(0, t_{\mathrm{unload},b_{k+j}} - t_{\mathrm{load_plane},b_{k+j}}) + \\
&\quad \lambda_1 \max(0, t_{\mathrm{load_plane},b_{k+j}} - \theta_{\mathrm{max_storage},b_{k+j}} - t_{\mathrm{unload},b_{k+j}})
\end{aligned}$$

where

- $t_{\text{unload},b_{k+j}}$ is the time instant when bag b_{k+j} is unloaded at its corresponding end point;
- $t_{\text{load\_plane},b_{k+j}}$ is the time instant when the end point closes;
- $\sigma_{b_{k+j}} \geq 1$ is the static priority of bag b_{k+j} (e.g. a measure of the priority of the flight);
- $\theta_{\text{max\_storage},b_{k+j}}$ is the maximum possible length of the time window for which the end point of bag b_{k+j} is open for that specific flight;
- $\lambda_1 \leq 1$ is a weighting parameter that represents the relative cost between buying additional storage space at the end points and the cost of customers that have their baggage delayed.

Note that the above performance function has some flat parts, which yields difficulties for optimization algorithms that use gradient information. To get some additional gradient we also include the dwell time (the time that the bag spends on the track network), resulting in:

$$J_{b_{k+j}}(t_{\text{unload},b_{k+j}}) = J_{\text{pen},b_{k+j}}(t_{\text{unload},b_{k+j}}) + \lambda_2 t_{\text{dwell},b_{k+j}}$$

where λ_2 is a small weighting factor ($\lambda_2 < \lambda_1$).

Then the local objective function $J_{\text{DMPC},N}(\mathbf{u}(k))$ is defined as:

$$J_{\text{DMPC},N}(\mathbf{u}(k)) = \sum_{j=1}^{N} J_{b_{k+j}}(\hat{t}^*_{\text{unload},b_{k+j}})$$

where $\hat{t}^*_{\text{unload},b_{k+j}}$ is the predicted unloading time instant of bag b_{k+j}. Next we present how the predicted unloading time instant is computed.

3.4 Prediction model

Our local prediction model is an even driven model for the local system where for each outgoing link $l \in \{0,1\}$ we consider a fixed release rate during the prediction period. The computation of a fixed link release rate is required due to the fact that we use a local simulation as prediction. Next we present how we calculate the release rate of a link given the state of the local system at t_{crt}. Let n_l denote the number of DCVs that left the outgoing link l within the time window $[t_{\text{crt}} - \tau_{\text{q}}, t_{\text{crt}}]$ of length τ_{q} time units. Then, if[1] $t_{\text{crt}} \geq \tau_{\text{q}}$, the fixed release rate of link l that will be used during the entire prediction period at bag step k is given by $\zeta_l = \dfrac{n_l}{\tau_{\text{q}}}$. Otherwise the release rate is at its maximum.

Recall that $u_{\text{sw\_out}}(k+j)$ with $j \in \{1,2,\ldots,N\}$ represents the position of the switch-out when bag b_{k+j} will cross S_s, determining the next junction towards which bag b_{k+j} will travel. Let $S_{\text{next},l}$ where $l = u_{\text{sw\_out}}(k+j)$ denote the junction that bag b_{k+j} will cross next, and let $S_{\text{dest},b_{k+j}}$ denote the corresponding end point of bag b_{k+j}.

For each possible route $r \in \mathscr{R}_{\text{next},b_{k+j},l}$, where $\mathscr{R}_{\text{next},b_{k+j},l}$ is the set of routes from $S_{\text{next},l}$ to $S_{\text{dest},b_{k+j}}$, we predict the time when bag b_{k+j} will arrive at $S_{\text{dest},b_{k+j}}$ via route r as follows:

$$\hat{t}_{\text{unload},b_{k+j},l,r} = \hat{t}_{\text{cross},b_{k+j}} + \hat{\tau}_{\text{link}\,l,b_{k+j}} + \hat{\tau}_{\text{route}\,r} \tag{1}$$

where

- $\hat{t}_{\text{cross},b_{k+j}}$ is the predicted time instant (computed by the local prediction model) at which bag b_{k+j} crosses S_s.
- $\hat{\tau}_{\text{link}\,l,b_{k+j}}$ is the time we predict that bag b_{k+j} spends on link l out of S_s. For this estimation we take:
$$\hat{\tau}_{\text{link}\,l,b_{k+j}} = \max\left(\frac{d_{\text{link}\,l}}{v_{\max}}, \frac{N_{\text{DCV},b_{k+j},l}}{\zeta_l}\right)$$
where $d_{\text{link}\,l}$ is the length of link, $v_{\max}$ is the maximal speed of a DCV, and $N_{\text{DCV},b_{k+j},l}$ is the number of DCVs on the link at the time instant $\hat{t}_{\text{cross},b_{k+j}}$.
- $\hat{\tau}_{\text{route}\,r}$ is the average travel time on route $r \in \mathscr{R}_{\text{next},b_{k+j},l}$ for an average speed determined based on historical data.

Then the optimal predicted unloading time instant is defined as follows:

$$\hat{t}^*_{\text{unload},b_{k+j}} = \underset{\{\hat{t}_{\text{unload},b_{k+j},l,r} \mid r \in \mathscr{R}_{\text{next},b_{k+j},l}\}}{\arg\min} J_{b_{k+j}}(\hat{t}_{\text{unload},b_{k+j},l,r}).$$

3.5 Optimization problem

So, the decentralized MPC optimization problem at junction S_s and bag step k is defined as follows:

$$\min_{\mathbf{u}(k)} J_{\text{DMPC},N}(\mathbf{u}(k))$$

subject to
the local prediction model
operational constraints

Since the optimization problem above involves integer variables, to solve it one could use *integer* optimization algorithms such as *genetic* algorithms or *tabu search* see e.g. Reeves and Rowe (2002); Glover and Laguna (1997).

The main advantage of decentralized MPC consists in a smaller computation time than the one needed when using centralized control due to the fact that we now compute in parallel the solution of a smaller and simplified optimization problem.

4. HEURISTIC APPROACHES

In order to lower the computation time of solving the switch control problem even more, in this section we propose two decentralized heuristic approaches to control the route of each DCV. In contrast to decentralized MPC, the heuristic approaches use a prediction horizon $N = 1$. Each switch is now locally controlled based on heuristic rules as presented next. We first consider the case where we determine the local switch control based only on local information regarding the flow of DCVs on the incoming and outgoing links of a junction. Let this junction be called S_s with $s \in \{1,2,\ldots,S\}$. Later on we also consider the case where additional data is used viz. information regarding the flow of DCVs on the incoming and outgoing links of the neighboring junctions of S_s.

For the sake of simplicity of notation, we will not explicitly include the subscript s in the remainder of this section since we describe the control of the switch-in and switch-out for one junction only.

4.1 Local information only

We now consider that the switch control is performed only based on local information regarding the flow of DCVs on the incoming and outgoing links of S_s.

[1] We assume the initial time of the simulation to be equal to 0 time units.

Control of the switch-in For a junction S_s, we define the following variables:

- Γ_l is the set of bags transported by DCVs that travel on the incoming link $l \in \{0,1\}$ of junction S_s at the time instant $t_{\text{enter},l}$ when a new bag enters the incoming link l;
- ρ_l^{static} is total static priority of link l, $\rho_l^{\text{static}} = \sum_{i \in \Gamma_l} \sigma_i$;
- ρ_l^{dyn} is the total dynamic priority of link l, $\rho_l^{\text{dyn}} = \sum_{i \in \Gamma_l} \frac{\hat{\delta}_i}{\delta_{\max,i}}$ with $\hat{\delta}_i$ the estimate of the actual time bag i requires to get from its current position to its final destination in case of no congestion and maximum speed, and $\delta_{\max,i}$ the maximum time left to bag i to spend in the system while still arriving at the plane on time. If bag i misses the flight, then the bag has to wait for a new plane with the same destination. Hence, a new departure time is assigned to bag i, and consequently a new loading time $t_{\text{new\_load\_plane},i}$ for bag i is considered. Then the variable $\delta_{\max,i}$ is defined as $\delta_{\max,i} = t_{\text{load\_plane},i} - t_{\text{enter},l}$ if $t_{\text{load\_plane},i} - t_{\text{crt}} > 0$ and $\delta_{\max,i} = t_{\text{new\_load\_plane},i} - t_{\text{enter},l}$ if $t_{\text{load\_plane},i} - t_{\text{enter},l} \leq 0$.

In order to determine the next position of the switch-in at junction S_s we compute the performance measure $p_{\text{sw\_in},l}$ for $l = 0,1$ every time a new bag enters the incoming link l. This performance measure takes into account the static and dynamic priorities of the bags transported by DCVs on the link l, and the current position of the switch-in at junction S_s (due to the operational constraint according to which the position of a switch at a junction can only change after minimum τ_{x} time units):

$$p_{\text{sw\_in},0} = w_{\text{st\_pr}}\rho_0^{\text{static}} + w_{\text{dyn\_pr}}\rho_0^{\text{dyn}} - w_{\text{sw\_in}}\tau_{\text{x}} I_{\text{crt}}$$
$$p_{\text{sw\_in},1} = w_{\text{st\_pr}}\rho_1^{\text{static}} + w_{\text{dyn\_pr}}\rho_1^{\text{dyn}} - w_{\text{sw\_in}}\tau_{\text{x}}(1 - I_{\text{crt}})$$

where I_{crt} denotes the current position of the switch-in at junction S_s (i.e. $I_{\text{crt}} = 0$ if the switch-in is positioned on the incoming link 0 and $I_{\text{crt}} = 1$ if the switch-in is positioned on the incoming link 1). The weighting parameters $w_{\text{st\_pr}}$, $w_{\text{dyn\_pr}}$, and $w_{\text{sw\_in}}$ can be calibrated as explained in Section 4.3.

Let $z_l \in \Gamma_l$ denote the bag closest to S_s on the incoming link l. The variable d_{z_l} denotes the distance between the current position of bag z_l and S_s and v_{z_l} denotes the current speed of the DCV transporting bag z_l. Then we define the time period $\tau_{\text{arrival\_at\_S}_\text{s},l}$ that the DCV transporting bag z_l needs to travel the distance d_{z_l} in case of no speed-update event as $\tau_{\text{arrival\_at\_S}_\text{s},l} = \frac{d_{z_l}}{v_{z_l}}$ if $d_{z_l} > 0$, and $\tau_{\text{arrival\_at\_S}_\text{s},l} = 0$ if $d_{z_l} = 0$.

The position of the switch-in at S_s is toggled only if $p_{\text{sw\_in},0} > p_{\text{sw\_in},1}$ and $I_{\text{crt}} = 1$, or if $p_{\text{sw\_in},1} > p_{\text{sw\_in},0}$ and $I_{\text{crt}} = 0$. If a toggle has to take place, then the switch-in changes position after $\tau_{\text{sw\_in},s} = \max(\tau_{\text{x}} - \tau_{\text{sw\_in\_prev},s}, \tau_{\text{arrival\_at\_S}_\text{s},1-I_{\text{crt}}})$ time units where $\tau_{\text{sw\_in\_prev},s}$ is the time for which the switch-in at junction S_s has been in its current position.

Control of the switch-out Every time when a bag is at junction S_s we compute the variable $\tau_{\text{sw\_out}}$ which represents the time period until the position of the switch-out has to be changed. This goes as follows.

Assume that bag i is at junction S_s. Then, using (1), we can predict the arrival time $\hat{t}_{\text{unload},i,l,r}$ of bag i at its corresponding end point $S_{\text{dest},i}$, when traveling on link $l \in \{0,1\}$ out of S_s and route $r \in \mathscr{R}_{\text{next},i,l}$ where $\mathscr{R}_{\text{next},i,l}$ is the set of routes from $S_{\text{next},l}$ to $S_{\text{dest},i}$.

Next we compute the cost criterion $c_{\text{sw\_out},i,l}$ for $l = 0,1$ that takes into account $J_i(\hat{t}^*_{\text{unload},i,l})$, where

$$\hat{t}^*_{\text{unload},i,l} = \underset{\{\hat{t}_{\text{unload},i,l,r} | r \in \mathscr{R}_{\text{next},i,l}\}}{\arg\min} J_i(\hat{t}_{\text{unload},i,l,r}),$$

and the current position O_{crt} of the outgoing switch:

$$c_{\text{sw\_out},i,0} = w_{\text{pen}} J_i(\hat{t}^*_{\text{unload},i,0}) + w_{\text{sw\_out}}\tau_{\text{x}} O_{\text{crt}}$$
$$c_{\text{sw\_out},i,1} = w_{\text{pen}} J_i(\hat{t}^*_{\text{unload},i,1}) + w_{\text{sw\_out}}\tau_{\text{x}}(1 - O_{\text{crt}}).$$

The weighting parameters w_{pen} and $w_{\text{sw\_out}}$ can be calibrated as explained in the Section 4.3.

The position of the switch-in at junction S_s is then toggled only if $c_{\text{sw\_out},i,0} < c_{\text{sw\_out},i,1}$ and $O_{\text{crt}} = 1$, or if $c_{\text{sw\_out},i,1} < c_{\text{sw\_out},i,0}$ and $O_{\text{crt}} = 0$. If the toggle has to take place, then the current position of the switch-out is changed after $\tau_{\text{sw\_out}} = \max(0, \tau_{\text{x}} - \tau_{\text{sw\_out\_prev}})$ where $\tau_{\text{sw\_out\_prev},s}$ is the time for which the switch-out at junction S_s has been in its current position.

4.2 Additional information from neighbors

In the sequel of this section we develop an approach where the switch control is performed based on both local information and additional data regarding the flow of DCV on the incoming and outgoing links of the neighboring junctions. This is an extension of 4.1.

Control of the switch-in Additionally to the variables defined in 4.1, for the junction S_s we define:

- Ω_l is the set of bags transported at the time instant $t_{\text{enter},l}$ on the incoming links of the neighboring junction $S_{\text{prev},l}$ that is linked to S_s via the incoming link l of S_s;
- $\varphi_l^{\text{static}}$ is total static priority of the incoming links of junction $S_{\text{prev},l}$, defined as $c_l = \sum_{i \in \Omega_l} \sigma_i$;
- φ_l^{dyn} is the total dynamic priority of the incoming links of $S_{\text{prev},l}$, defined as $d_l = \sum_{i \in \Omega_l} \frac{\hat{\delta}_i}{\delta_{\max,i}}$.

The time when junction S_s toggles its position is computed as in the paragraph *Control of the switch-in* of Section 4.1. The difference is that here we use the following performance measures:

$$p_{\text{sw\_in},0} = w_{\text{st\_pr}}\rho_0^{\text{static}} + w_{\text{dyn\_pr}}\rho_0^{\text{dyn}} + w_{\text{ad}}(w_{\text{st\_pr}}\varphi_0^{\text{static}} + w_{\text{dyn\_pr}}\varphi_0^{\text{dyn}}) - w_{\text{sw\_in}}\tau_{\text{x}} I_{\text{crt}}$$
$$p_{\text{sw\_in},1} = w_{\text{st\_pr}}\rho_1^{\text{static}} + w_{\text{dyn\_pr}}\rho_1^{\text{dyn}} + w_{\text{ad}}(w_{\text{st\_pr}}\varphi_1^{\text{static}} + w_{\text{dyn\_pr}}\varphi_1^{\text{dyn}}) - w_{\text{sw\_in}}\tau_{\text{x}}(1 - I_{\text{crt}}).$$

The additional weighting parameter w_{ad} represents the influence of the additional information on the performance index $p_{\text{sw\_in},l}$, for $l = 0,1$. This weighting parameter is calibrated as explained in Subsection 4.3.

Control of the switch-out The position of the switch-out of junction S_s is computed similarly to Section 4.1. However, in

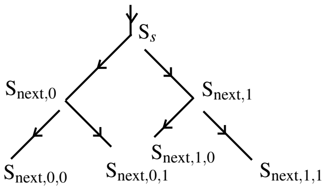

Fig. 4. Relevant neighbors of S_s.

this case, when computing the predicted objective function for link $l = 0,1$ and bag i, we do not look only at the congestion on the outgoing links of junction S_s, but also at the congestion on the outgoing links of $S_{next,l}$ (where $S_{next,l}$ is the neighboring junction of S_s connected via the outgoing link l of S_s).

As sketched in Figure 4, let $S_{next,l,m}$ for $m = 0,1$ denote the neighboring junction of $S_{next,l}$ connected via link m out of $S_{next,l}$. Also, let $N_{new\,DCV,i,l,m}$ denote the number of DCVs on link l out of S_s that will choose link m out of $S_{next,l}$. We assume that for a junction $S_{next,l}$, $l \in \{0,1\}$ with 2 outgoing links, half of the DCVs traveling from S_s to $S_{next,l}$ take link $m = 0$ out of $S_{next,l}$, and the other half take link $m = 1$. Then $\hat{\tau}_{link\,l,m,i} = \max\left(\frac{d_{link\,l,m}}{v_{max}}, \frac{N_{DCV,i,l,m} + N_{new\,DCV,i,l,m}}{\zeta_{l,m}}\right)$ is the time period that bag i needs to travel link m out of $S_{next,l}$ considering the release rate $\zeta_{l,m}$ of link m out of $S_{next,l}$, where $d_{link\,l,m}$ is the length of the link m out of $S_{next,l}$ and $N_{DCV,i,l,m}$ is the number of DCVs on this link at the time instant when bag i crosses junction S_s.

Let $\mathscr{R}_{next,i,l,m}$ with $l \in \{0,1\}$ and $m \in \{0,1\}$ denote the set of routes from junction $S_{next,l,m}$ to $S_{dest,i}$. In this case, for each route $r \in \mathscr{R}_{next,i,l,m}$ we predict the time $\hat{t}_{unload,i,l,m,r}$ when bag i will reach $S_{dest,i}$ if the bag takes link l out of S_s, link m out of $S_{next,l}$, and route r. This time is given by:

$$\hat{t}_{unload,i,l,m,r} = t_{cross\,S_s,i} + \hat{\tau}_{link\,l,i} + \hat{\tau}_{link\,l,m,i} + \hat{\tau}_{route\,r}$$

where $t_{cross\,S_s,i}$ is the time instant when bag i crosses junction S_s, $\hat{\tau}_{travel\,link\,l,i}$ is the time we predict that bag i will spend on link l out of S_s, and $\hat{\tau}_{route\,r}$ is the average travel time on route r for an average speed empirically determined.

Finally, in computing the cost criterion $c_{sw\_out,i,l}$ for $l = 0,1$ defined in Section 4.1 we use $J_i(\hat{t}^*_{unload,i,l})$ where $\hat{t}^*_{unload,i,l}$ is the predicted unloading time that optimizes the performance index of bag i when choosing link $m \in \{0,1\}$ out of $S_{next,l}$, and route $r \in \mathscr{R}_{next,i,l,m}$:

$$\hat{t}^*_{unload,i,l} = \underset{\{\hat{t}_{unload,i,l,m,r} | m \in \{0,1\} \wedge r \in \mathscr{R}_{next,i,l,m}\}}{\arg\min} J_i(\hat{t}_{unload,i,l,m,r}).$$

4.3 Calibration

The calibration of the weighting parameters presented in the Sections 4.1 and 4.2 will be done by solving the following optimization problem for a set of typical scenarios:

$$\min_{\mathbf{w}} \sum_{j=1}^{N_{scenario}} J_{j,tot}(\mathbf{w})$$

subject to

the system dynamics and control actions depending on $\mathbf{w}$

operational constraints

where $\mathbf{w}$ is $[w_{st\_pr}\, w_{dyn\_pr}\, w_{pen}\, w_{sw\_in}\, w_{sw\_out}]^T$ for 4.1, or $[w_{st\_pr}\, w_{dyn\_pr}\, w_{pen}\, w_{sw\_in}\, w_{sw\_out}\, w_{ad}]^T$ for 4.2, $N_{scenario}$ is the number of scenarios over which the calibration is performed, and $J_{j,tot}$ is the total performance index corresponding to scenario j, $j \in \{1,2,\ldots,N_{scenario}\}$. The total performance index of the DCV-based baggage handling system for a given scenario is defined as $J_{tot} = \sum_{i \in \Lambda} J_{pen,i}(t_{unload,i})$, where Λ is the set of bags to be handled during the entire simulation.

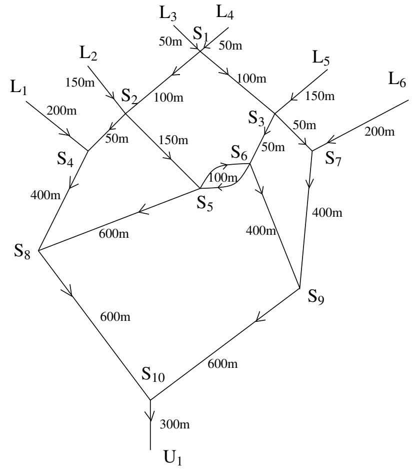

Fig. 5. Case study for a DCV-based baggage handling system.

The above optimization problem is nonlinear and nonconvex, and has continuous variables. So, in order to solve this problem, one could use global optimization algorithms such as multi-start *sequential quadratic programming*, *pattern search*, *simulated annealing* algorithms, or multi-start *genetic* algorithms, see e.g. (Pardalos and Resende, 2002).

5. CASE STUDY

In this section we compare the proposed control methods based on a simulation example.

5.1 Set-up

We consider the network of tracks depicted in Figure 5 with 6 loading stations, 1 unloading station, and 10 junctions. 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 $v_{max} = 20$ m/s, being controlled by on-board collision avoidance controllers. The lengths of the track segments are indicated in Figure 5.

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.

5.2 Scenarios for calibration and control

For the calibration of the weighting parameters we have defined 27 scenarios where 180 bags have to be handled.

We have considered typical scenarios with different classes of demand profiles for each loading station, different initial states of the system, queues on different links, and different time criticality measures (e.g. cases where the transportation of the bags is very tight, i.e. the last bag that enters the system can only

Table 1. Comparison of average performance of the system and computation time for the proposed control methods.

control approach	$J_{\text{avg,control approach}}$ (s)	CPU time (s)
CMPC ($N = 5$)	$1.79 \cdot 10^3$	$1.59 \cdot 10^5$
DMPC ($N = 5$)	$2.33 \cdot 10^3$	$2.95 \cdot 10^3$
HR (local)	$4.32 \cdot 10^3$	0.99
HR (more info)	$3.46 \cdot 10^3$	2.17

arrive in time at the corresponding end point if the shortest path is used and its DCV is continuously running with maximum speed, or cases where the timing is more relaxed).

For comparing the control methods we have used the same scenarios, but different samples of the demand profiles than those considered for calibrating the weighting parameters **w**.

5.3 Results

To solve the local optimization problem of decentralized MPC and the calibration problem of the heuristic approaches we have chosen a *genetic* algorithm with multiple runs since experiments indicate that this optimization technique gives good performance, with the shortest computation time.

Based on simulations we now compare, for the same scenarios, the proposed control methods. Let $J_{j,\text{control approach}}$ denote the performance index of the baggage handling system corresponding to scenario index j and the considered control approach.

In Table 1 we list the average results

$$J_{\text{avg,control approach}} = \frac{1}{N_{\text{scenario}}} \sum_{j=1}^{N_{\text{scenario}}} J_{j,\text{control approach}}$$

obtained when using centralized MPC, decentralized MPC, and the decentralized heuristic approaches, where N_{scenario} is the number of considered scenarios.

The results indicate that decentralized MPC involves a good trade-off between computation time [2] and optimality, the performance being influenced by the considered horizon. In Figure 6 we illustrate the dependence of the performance and computation time upon the prediction horizon N when using decentralized MPC for a typical scenario. The simulations show that by increasing the horizon the performance typically improves, but at the cost of higher computation time. Moreover, the heuristic approaches perform very fast, but, usually, the results are worse than those obtained when using decentralized MPC.

6. CONCLUSIONS AND FUTURE WORK

In this paper we have considered baggage handling processes using destination coded vehicles (DCVs) that transport bags at high speed on a "mini" railway network. In particular we have considered the route choice control problem for each DCV transporting bags on the track network. The best performance of the system is obtained when using centralized switch control. However, in practice, this approach is not tractable due to the very high computational effort that centralized control methods require to solve an optimization problem that is nonlinear, nonconvex, and with integer valued variables. Therefore, in this paper, we have developed and compared approaches to control the switches of the network (and implicitly the route for each DCV transporting bags) in a decentralized way. Three approaches have been proposed viz. decentralized model predictive control (MPC) and two heuristic approaches. Experiments indicate that decentralized MPC can offer a good trade-off between the optimality and the time required to compute the route for each DCV. On the other hand, the heuristic approaches perform very fast, but usually, the results are worse than those obtained when using decentralized MPC.

In future work we will include coordination between the neighboring junctions, and assess the benefits that can be obtained by using such distributed control.

[2] The simulations were performed on a 3.0 GHz P4 with 1 GB RAM.

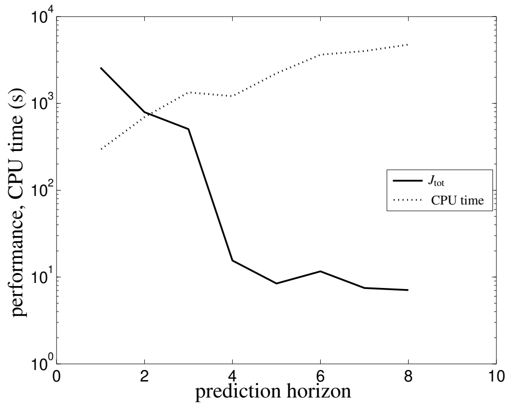

Fig. 6. Dependence of the performance and CPU time upon the prediction horizon for DMPC, for a typical scenario with non tight timing.

ACKNOWLEDGMENTS

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

REFERENCES

Gang, H., Shang, J., and Vargas, L. (1996). A neural network model for the free-ranging AGV route-planning problem. *Journal of Intelligent Manufacturing*, 7(3), 217–227.

Glover, F. and Laguna, F. (1997). *Tabu Search*. Kluwer Academic Publishers, Norwell, Massachusetts, USA.

Kaufman, D., Nonis, J., and Smith, R. (1998). A mixed integer linear programming model for dynamic route guidance. *Transportation Research Part B: Methodological*, 32(6), 431–440.

Maciejowski, J. (2002). *Predictive Control with Constraints*. Prentice Hall, Harlow, UK.

Pardalos, P. and Resende, M. (eds.) (2002). *Handbook of Applied Optimization*. Oxford University Press, Oxford, UK.

Reeves, C. and Rowe, J. (2002). *Genetic Algorithms - Principles and Perspectives: A Guide to GA Theory*. Kluwer Academic Publishers, Norwell, Massachusetts, USA.

Tarău, A., De Schutter, B., and Hellendoorn, J. (2008). Travel time control of destination coded vehicles in baggage handling systems. In *In the IEEE International Conference on Control Applications*, 293–298. San Antonio, Texas, USA.