Delft University of Technology

Delft Center for Systems and Control

Technical report 08-030

Model-based control for throughput optimization of automated flats sorting machines*

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, "Model-based control for throughput optimization of automated flats sorting machines," *Control Engineering Practice*, vol. 17, no. 6, pp. 733–739, June 2009. doi:10.1016/j.conengprac.2008.11.007

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

Model-based control for throughput optimization of automated flats sorting machines

A.N. Tarău [a,*] B. De Schutter [a,1] J. Hellendoorn [a]

[a] *Delft University of Technology, Delft Center for Systems and Control, Mekelweg 2, 2628 CD Delft, The Netherlands*

Abstract

Mail items of A4 size are called flats. In order to handle the large volumes of flats that have to be processed, state-of-the-art post sorting centers are equipped with dedicated flats sorting machines. The throughput of a flats sorting machine is crucial when dealing with a continually increasing number of items to be sorted in a certain time. But, the throughput is limited by the mechanical constraints. In order to optimize the efficiency of this sorting system, in this paper, several design changes are proposed and advanced model-based control methods such as optimal control and model predictive control are implemented. An event-based model of the flats sorting system is also determined using simulation. The considered control methods are compared for several scenarios. The results indicate that by using the proposed approaches the throughput can be increased with over 20%.

Key words: Transportation systems, throughput optimization, speed control, discrete-event simulation.

1 Introduction

During the last decades the volume of magazines, catalogs, and other plastic wrapped mail items that have to be processed by post sorting centers has increased considerably. In order to be able to handle the large volumes of mail state-of-the-art post sorting centers are equipped with dedicated mail sorting

* Corresponding author. Tel.: +31 15 27 82087; fax: +31 1527 86679.
Email address: a.n.tarau@tudelft.nl (A.N. Tarău).

[1] B. De Schutter is also with the Marine and Transport Technology department of the Delft University of Technology.

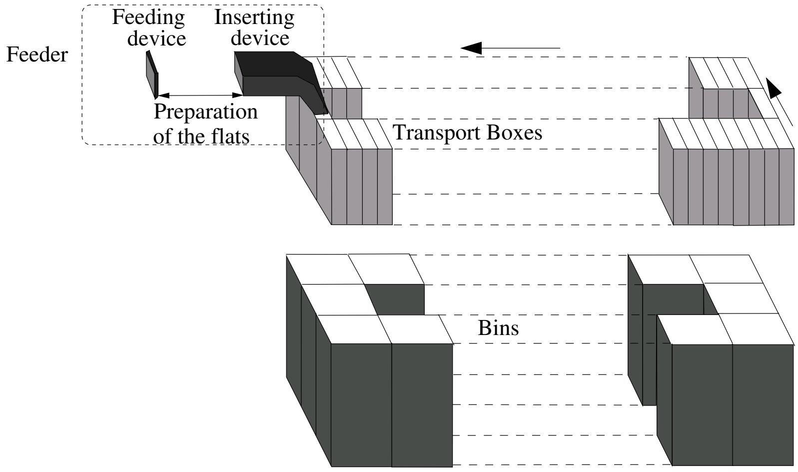

Fig. 1. Sorting part of a flats sorting machine.

machines. Nowadays, the focus is on quality, reliability, and throughput maximization. The throughput of a post sorting machine is defined as the number of sorted mail items divided by the time needed to sort them. The throughput is limited by the mechanical constraints and also by the performance of the address reading devices. In this paper large letters of A4 size envelopes are considered. Such mail items are called flats.

The procedure performed by a flats sorting machine consists of two processes: preparing the flats and sorting them. During the preparation phase, the flats are faced in the same way, and the stamp used for postage is voided. Next, the address and the postal code are located and the necessary information is extracted and printed on the flat in form of a bar code. Conveyor systems transport the flats during the preparation phase with a constant speed. The length of these conveyor belts and their speed determine the maximal amount of time available to prepare the mail for sorting. If there is non machine-readable information, an identification code is assigned to the item and the flat is sent into a special bin for manual processing. When the mail item leaves the preparation phase, it is inserted into a transport box of the sorting process by the inserting device, as illustrated in Figure 1. The transport box carries the flat and deposits it by dropping it into a destination bin according to the destination or postal code of the flat. This is how currently most of the flats sorting machines are working.

The low-level control problems of this system consist of setting the feeding rate of the sorting machine Lohmann (1996), positioning of the transport box when inserting the flat, and synchronizing transport boxes and bins when dropping a flat in its corresponding destination bin. At a higher level of control an important problem is how to allocate the destinations to the bins.

Topics such as locating the destination address and extracting the necessary information, and also designing optical character recognition machines have been treated to a very large extent in e.g. Kaneko and Fujiwara (2004), Leimer (1962), Whichello and Yan (1996). In this paper approaches to increase the throughput of the flats sorting machine are investigated. This can be achieved first by making minor design changes such as augmenting the system with additional feeders and also by moving the bin system to the left or to the right with a given speed. Advanced model-based control methods will be implemented in order to ensure the optimal movements. The control approaches considered in this paper are optimal control and model predictive control.

The paper is organized as follows. In Section 2, the proposed new set-up of the flats sorting machine is described. The simplifying assumptions and the continuous-time event-driven model to be used are presented in Section 3. In Section 4, several control approaches are proposed for determining the velocity of the system transporting the bins. The analysis of the simulation results and the comparison of the proposed control methods are elaborated in Section 5. In Section 6, the influence of the structural changes on the throughput is discussed. Finally, in Section 7, the conclusions are drawn and possible directions for future research are presented.

2 Automated flats processing – New set-up

In order to increase the throughput of the flats sorting machine, the new set-up illustrated in Figure 3 is proposed.

The preparation of the flats is identical to the one described in Section 1. But, in order to simplify the previous explanation, instead of the feeding device and the preparation phase, a buffer of flats the codes of which are known in advance will be considered.

The sorting system is augmented by adding feeders, which can increase the throughput. However, by increasing the number of feeders only, one does not necessarily obtain the maximal possible throughput. From Figure 2, which shows the throughput versus the velocity of the bin system for a typical scenario (see Section 5.1), one concludes that the throughput obtained with a static bin system is not optimal. Therefore, in the new set-up, the bottom system transporting the bins is also able to move to the left, to the right, or not at all. Moving the bottom part of the flats sorting machine, but with a constant speed, is currently already being implemented in prototypes. The top system transporting the boxes moves as usual, with a constant speed. The reason for this is to increase the number of empty transport cassettes and, hence, the availability of the transport boxes.

Consequently, a new control problem arises: how to adjust the speed of the

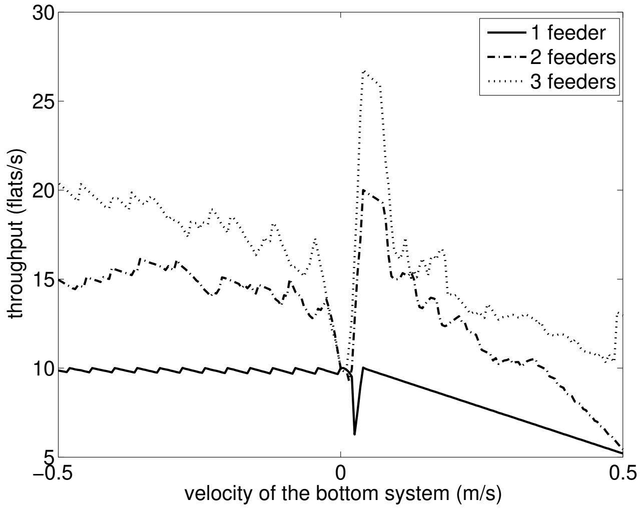

Fig. 2. Throughput versus v_{bottom} when $v_{\mathrm{top}} = 1\,\mathrm{m/s}$ and v_{bottom} varies between e.g. $-0.5\,\mathrm{m/s}$ and $0.5\,\mathrm{m/s}$ (v_{bottom} is assumed negative when the bottom system has the opposite sense of movement with respect to the top system).

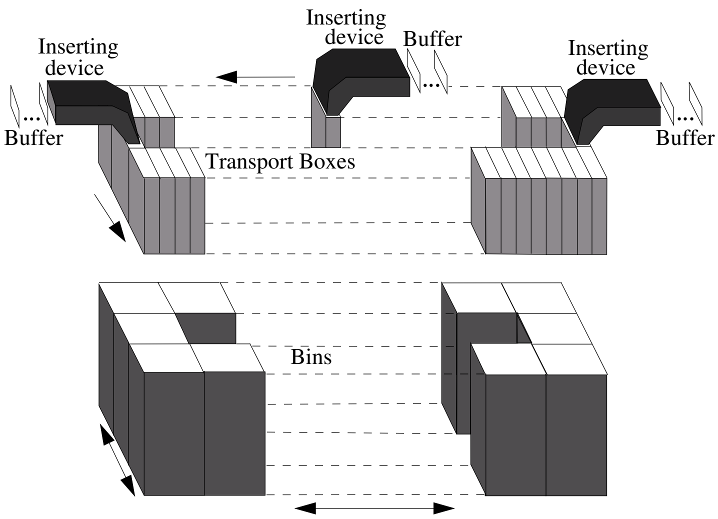

Fig. 3. New set-up for the flats sorting machine.

bottom system, so that the throughput is maximized.

3 Assumptions and chosen model

Consider the simplified process depicted in Figure 3 of a flats sorting system with F feeders. Accordingly, F FIFO (First In First Out) buffers of flats are fed into the system.

Later on the model will be used for on-line model-based control. So, in order to obtain a trade-off between a detailed model that requires large computation

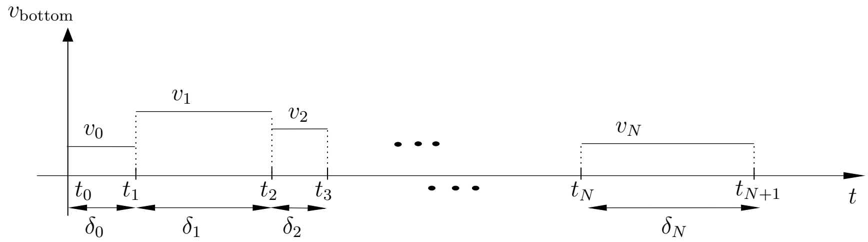

Fig. 4. Speed evolution of the bin system.

time and a fast simulation the following assumptions were made:

A_1: The top system moves with a constant speed v_{top}.
A_2: The speed of the bottom system is piecewise constant.
A_3: The flats sorting machine has F inserting devices which are positioned equidistantly.
A_4: Each inserting device has a finite buffer of flats, with codes that are known in advance. This allows to compute the optimal speed profile in advance.
A_5: When using a flats sorting machine with F feeders, the stream of codes $\mathbf{s} = [s_1\ \ s_2 \ldots\ s_f]^{\mathrm{T}}$ where f is the number of flats to be sorted during a sorting round, is split in F new streams $\mathbf{s}_1 = [s_1\ \ s_2 \ldots s_l]^{\mathrm{T}}$, $\mathbf{s}_2 = [s_{l+1}\ \ s_{l+2} \ldots s_{2l}]^{\mathrm{T}}, \ldots,\ \mathbf{s}_F = [s_{(F-1)l+1}\ \ s_{(F-1)l+2} \ldots s_f]^{\mathrm{T}}$ with $l = \left\lfloor \frac{f}{F} \right\rfloor$, where $\lfloor x \rfloor$ denotes the largest integer less than or equal to x.
A_6: The correct dropping (and in consequence the correct stacking) of a flat in a bin is controlled by a low-level controller.
A_7: A full bin is replaced with a new one in a negligible time span.

There are three types of events that can occur:

- inserting a new flat into the sorting section of the system,
- dropping the flats that meet the corresponding bin,
- updating the speed of the bottom system.

The flats sorting system is modeled as an event-driven model consisting of a continuous part, viz. the movement of the transport boxes and bins, and of the discrete events listed above. The following situation has been assumed: given a velocity sequence $\mathbf{v} = [v_0\ v_1 \ldots v_N]^{\mathrm{T}}$ and a sequence of time interval lengths $\boldsymbol{\delta} = [\delta_0\ \delta_1 \ldots \delta_N]^{\mathrm{T}}$, on each time interval $[t_k, t_{k+1})$, $k = 0, 1, \ldots, N$, with $t_{k+1} = t_k + \delta_k$ and t_0 the initial time, the velocity of the bottom system equals v_k as illustrated in Figure 4.

The model of the flats sorting system is captured by Algorithm 1, where the inserting devices are denoted by F_1, F_2, ..., F_F, and where $T \geq 0$ is the sorting time for the entire stream of f flats that enter the system.

According to the model, for each flat i, with $i = 1, 2, \ldots, f$ that has to be sorted, the time instant when the flat i is inserted into a box ($t_{\text{insert},i}$) and the time instant when the flat i is dropped ($t_{\mathrm{drop},i}$) are computed. Consequently, the model of the flats sorting machine is denoted by $\mathbf{t} = \mathcal{M}(\mathbf{v}, \boldsymbol{\delta})$, where $\mathbf{t} = [t_{\mathrm{insert},1} \dots t_{\mathrm{insert},f}\ t_{\mathrm{drop},1}\ \dots\ t_{\mathrm{drop},f}]^{\mathrm{T}}$.

Algorithm 1. Flats sorting

$k \leftarrow 0$
$t \leftarrow t_0$
$\mathbf{v} \leftarrow [v_0\ v_1 \dots v_N]^{\mathrm{T}}$
$\boldsymbol{\delta} \leftarrow [\delta_0\ \delta_1 \dots \delta_N]^{\mathrm{T}}$
while $t \leq t_0 + T$ **do**
  for $j = 1$ to F **do**
    $\delta_{\mathrm{insert},j} \leftarrow$ time that will pass until the next inserting event from F_j's point of view
  end for
  $\delta_{\mathrm{drop}} \leftarrow$ time that will pass until the first next dropping event
  $\delta_{\min} \leftarrow \min(\min\limits_{j=1,\dots,F} \delta_{\mathrm{insert},j}, \delta_{\mathrm{drop}}, \delta_k)$
  $t \leftarrow t + \delta_{\min}$
  $p_{\mathrm{box}} \leftarrow p_{\mathrm{box}} + v_{\mathrm{top}}\delta_{\min}$ {update the position of the box system}
  $p_{\mathrm{bin}} \leftarrow p_{\mathrm{bin}} + v_k\delta_{\min}$ {update the position of the bin system}
  $\Gamma \leftarrow$ set of flats to be inserted in the boxes next
  $\Delta \leftarrow$ set of flats to be dropped next
  for all $j \in \Gamma$ **do**
    $t_{\mathrm{insert},j} \leftarrow t$
  end for
  for all $j \in \Delta$ **do**
    $t_{\mathrm{drop},j} \leftarrow t$
  end for
  $\delta_k \leftarrow \delta_k - \delta_{\min}$
  if $\delta_k = 0$ **then**
    $k \leftarrow k + 1$
  end if
end while

The operational constraints derived from the mechanical and design limitations of the machine are the following:

- the velocities of the bottom and top system are bounded,
- $\delta_k \geq \delta_{\mathrm{x}}$ for $k = 0, 1, \dots, N$ with δ_{x} the minimum time period for which the velocity of the bottom system stays constant.

These constraints are denoted by $\mathcal{C}(\mathbf{v}, \boldsymbol{\delta}, \delta_{\mathrm{x}}) \leq 0$.

4 Control approaches

In order to increase the throughput of the flats sorting system, several model-based control approaches with various degrees of complexity are proposed to determine the speed of the bottom system.

4.1 Optimal control with a piecewise constant speed on time intervals of variable length

Several methods for solving dynamic optimization problems have been developed. The optimal control problem consists of finding the time-varying control law $u(\cdot)$ for a given system such that a performance index J is optimized while satisfying the operational constraints imposed by the model, see e.g. Lewis (1986). However, continuous adjustment of the velocity of the bottom system so as to maximize the throughput J of a flats sorting machine is not possible using digital control. Hence, the assumption that v_{bottom} is piecewise constant is necessary.

One may divide the period $[t_0, t_0 + T]$ into $N + 1$ time intervals of variable length δ_0, $\delta_1, \ldots, \delta_N$ such that $\sum_{k=0}^{N} \delta_k = T$. Define the piecewise constant control law $u_{\mathrm{pwc}} : \{0, 1, \ldots, N\} \to \mathbb{R}$ and the time instants $t_{k+1} = t_k + \delta_k$. Then the piecewise constant speed u_{pwc}, and the intervals δ_0, $\delta_1, \ldots, \delta_N$ that maximize J have to be computed. The optimal control problem is defined as follows:

$$
\mathrm{P}_1: \quad \max_{u_{\mathrm{pwc}}, \boldsymbol{\delta}} J(u_{\mathrm{pwc}}, \boldsymbol{\delta})
$$

subject to

$$
\mathbf{t} = \mathcal{M}(\mathbf{v}, \boldsymbol{\delta}) \text{ and } \mathcal{C}(\mathbf{v}, \boldsymbol{\delta}, \delta_{\mathrm{x}}) \leq 0
$$

where $\mathbf{v} = [u_{\mathrm{pwc}}(0) \;\; u_{\mathrm{pwc}}(1) \ldots u_{\mathrm{pwc}}(N)]^{\mathrm{T}}$ is the control sequence and $\boldsymbol{\delta} = [\delta_0 \; \delta_1 \ldots \delta_N]^{\mathrm{T}}$ is the corresponding sequence of variable time interval lengths.

4.2 Optimal control with a piecewise constant speed on time intervals of constant length

One may further simplify the problem P_1 by considering δ_k equal to a sampling time T_{s} for $k = 0, 1, \ldots, N$. Accordingly, this optimal control problem is defined as follows:

$$
\mathrm{P}_2: \quad \max_{u_{\mathrm{pwc}}} J(u_{\mathrm{pwc}})
$$

subject to

$$\mathbf{t} = \mathcal{M}(\mathbf{v}, \boldsymbol{\delta}) \text{ and } \mathcal{C}(\mathbf{v}, \boldsymbol{\delta}, \delta_{\mathrm{x}}) \leq 0$$

where $\mathbf{v} = [u_{\mathrm{pwc}}(0)\ u_{\mathrm{pwc}}(1) \ldots u_{\mathrm{pwc}}(N)]^{\mathrm{T}}$ and $\boldsymbol{\delta} = [T_{\mathrm{s}}\ T_{\mathrm{s}} \ldots T_{\mathrm{s}}]^{\mathrm{T}}$.

In both P_1 and P_2, the throughput increases monotonically with a smaller T_{s} or an increasing N until the best achievable throughput is reached. However, this comes at the cost of a higher computation time.

4.3 Optimal control with a constant speed

Now consider the simplest case of P_1 and P_2. For the entire stream of flats entering the system in one sorting round, the constant speed u_{ct} that maximizes the throughput is computed. This optimal control problem can be defined as follows:

$$\mathrm{P}_3\text{:} \quad \max_{u_{\mathrm{ct}}} J(u_{\mathrm{ct}})$$

subject to

$$\mathbf{t} = \mathcal{M}(\mathbf{v}, \boldsymbol{\delta}) \text{ and } \mathcal{C}(\mathbf{v}, \boldsymbol{\delta}, \delta_{\mathrm{x}}) \leq 0$$

where $\mathbf{v} = [u_{\mathrm{ct}}]$ and $\boldsymbol{\delta} = [T]$.

The throughput obtained by solving P_3 is in general smaller than the one obtained by using optimal control with variable speed, but the computation time also decreases significantly.

4.4 Model predictive control

In order to make a trade-off between the optimality of the throughput and the time required to compute the optimal velocity sequence of the bottom system, model predictive control (MPC) is introduced.

Model predictive control is an on-line control design method that uses the receding horizon principle, see e.g. Maciejowski (2002).

In this approach, given a prediction horizon N_{p} and a control horizon N_{c} with $N_{\mathrm{c}} \leq N_{\mathrm{p}}$, at step k, the future control sequence $u(0|k), \ldots, u(N_{\mathrm{c}} - 1|k)$ is computed by solving a discrete-time optimization problem over a given period $[t_k, t_k + N_{\mathrm{p}} T_{\mathrm{s}}]$ with $t_k = t_0 + k T_{\mathrm{s}}$, so that the cost criterion J is optimized subject to the operational constraints. The input signal is typically assumed to become constant beyond the control horizon i.e. $u(j|k) = u(N_{\mathrm{c}} - 1|k)$ for

$j \geq N_\mathrm{c}$. 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, and a new optimization problem at step $k+1$ is solved using this new information. In this way, a feedback mechanism is introduced.

The MPC optimization problem can be solved by using a modified version of optimal control with a piecewise constant speed where at step k, $u_\mathrm{pwc}(j) = u(j|k)$ and $\delta(j) = T_\mathrm{s}$ for $j = 0, 1 \ldots, N_\mathrm{c} - 1$.

This results in the following problem:

$$
\begin{aligned}
\mathrm{P}_4\colon\quad & \max_{u_\mathrm{pwc}} J(u_\mathrm{pwc}) \\
& \text{subject to} \\
& \quad \mathbf{t} = \mathcal{M}(\mathbf{v}, \boldsymbol{\delta}) \\
& \quad \mathcal{C}(\mathbf{v}, \boldsymbol{\delta}, \delta_\mathrm{x}) \leq 0 \\
& \quad u_\mathrm{pwc}(j) = u_\mathrm{pwc}(N_\mathrm{c} - 1) \text{ for } j \geq N_\mathrm{c}
\end{aligned}
$$

where $\mathbf{v} = [u_\mathrm{pwc}(0) \;\; u_\mathrm{pwc}(1) \ldots u_\mathrm{pwc}(N_\mathrm{c} - 1)]^\mathrm{T} \ldots u_\mathrm{pwc}(N_\mathrm{c} - 1)]^\mathrm{T}$ and $\boldsymbol{\delta} = [T_\mathrm{s}\; T_\mathrm{s} \ldots T_\mathrm{s}]^\mathrm{T}$.

Note that in this paper the prediction horizon is not fixed, but determined by using the following procedure[2]. Assume that one wants to determine in advance the optimal speed profile for sorting b flats. Then for $N_\mathrm{c}T_\mathrm{s}$ time units the velocity sequence $\mathbf{u}_\mathrm{pwc}$ is used. Each velocity $u_\mathrm{pwc}(j)$, with $j = 0, 1, \ldots, N_\mathrm{c} - 1$, is applied during a sampling period of length T_s. All this results in $b_1 \leq b$ flats being sorted in the period $[t_0, t_0 + N_\mathrm{c}T_\mathrm{s}]$. For sorting the rest of $b - b_1$ flats, the velocity of the bottom system is kept constant, equal to $u_\mathrm{pwc}(N_\mathrm{c} - 1)$. Consequently, the time T_horizon needed to sort the b flats divided by T_s determines the prediction horizon as follows: $N_\mathrm{p} = \left\lceil \frac{T_\mathrm{horizon}}{T_\mathrm{s}} \right\rceil$ where $\lceil x \rceil$ denotes the smallest integer larger than or equal to x.

The main advantage of MPC compared to optimal control with variable speed is given by a smaller computation time since $N_\mathrm{p}T_\mathrm{s} \leq T$. Even more, the velocity of the bottom system may be computed on-line. However, this happens at the cost of a suboptimal throughput.

[2] In this variant of MPC the prediction horizon corresponds to the number of flats to be sorted in advance. This is required in order to be able to compare the obtained throughput values for different control inputs in a correct way.

4.5 Optimization methods

In order to solve the optimization problems presented in the previous sections and hence, to determine the optimal speed of the bottom system of a flats sorting machine, one may use *sequential quadratic programming* algorithm, *pattern search* or *genetic* algorithms, see e.g. Gill et al. (1981), Audet and Dennis Jr. (2002), Goldberg (1989).

5 Simulation results

In this section the proposed control methods are compared based on simulation examples.

5.1 Scenarios

Recall that the velocity v_{top} of the top system is constant. Assume v_{top} to be equal to $1\,\text{m/s}$, while the velocity v_{bottom} of the bottom system varies between $-0.5\,\text{m/s}$ and $0.5\,\text{m/s}$. It is also assumed that the width of the bin is four times the width of the box. The examples in the following sections involve $N_{\text{bin}} = 100$ bins, respectively $N_{\text{box}} = 400$ boxes, while the length f of the stream of flats that enter the sorting system equals 24000.

Several scenarios are considered, where the stream consists of:

Scenario 1: perfectly ordered codes (i.e. with the same order as the order of codes allocated to the bins). Since the width of the bin is four times the width of the box, a perfectly ordered stream of flats is a stream of the form e.g. $1, 1, 1, 1, 2, 2, 2, 2, \ldots, N_{\text{bin}}, N_{\text{bin}}, N_{\text{bin}}, N_{\text{bin}}, 1, 1, 1, 1, \ldots$. The order of the N_{bin} bins passing under the first inserting device when the bottom system moves to the right is in this case $1, 2, 3, \ldots, N_{\text{bin}}$.

Scenario 2: alternating sequences of random, and respectively ordered codes e.g. $\boldsymbol{\sigma}_1, \boldsymbol{\sigma}_2, \ldots, \boldsymbol{\sigma}_m$ where if $\boldsymbol{\sigma}_j$ is a sequence of random codes, $1 \leq j < m$, then $\boldsymbol{\sigma}_{j+1}$ is a sequence of ordered codes. The length of each sequence $\boldsymbol{\sigma}_j$ for $j = 1, 2, \ldots, m$ is also chosen randomly. This scenario has been chosen due to the fact that the mail may be partially presorted.

Scenario 3: completely random codes.

For the scenarios above, the throughput of the flats sorting machine which has the bottom system static (i.e. $v_{\text{bottom}} \equiv 0\,\text{m/s}$) has been listed in Table 1.

Table 1
The throughput J_{opt} (flats/s) for the scenarios considered, when the bottom system of the sorting machine is static ($v_{bottom} = 0\,\mathrm{m/s}$).

Scenario	1 feeder	2 feeders
1	9.90	13.19
2	9.85	13.14
3	9.85	13.07

Table 2
Comparison of throughput and computation time obtained by using the Matlab functions *fmincon*, *patternsearch*, and *ga* for the set-up with two feeders.

	J_{opt} (flats/s)			computation time (s)		
Scen.	*fmincon*	*patternsearch*	*ga*	*fmincon*	*patternsearch*	*ga*
1	15.68	15.68	15.68	$2.39 \cdot 10^3$	$6.32 \cdot 10^2$	$1.74 \cdot 10^3$
2	15.81	15.84	15.84	$4.63 \cdot 10^3$	$1.20 \cdot 10^3$	$4.01 \cdot 10^3$
3	15.79	15.88	15.77	$5.27 \cdot 10^3$	$1.22 \cdot 10^3$	$4.20 \cdot 10^3$

5.2 Optimal control with a constant speed

In this section the constant speed of the bottom system that optimizes the throughput for all the flats that enter the system in one sorting round is computed. So, the optimization problem P_3 is solved.

Figure 2 shows the throughput versus velocity of the bottom system by discretizing the velocity with the sampling step of e.g. $0.005\,\mathrm{m/s}$. One may notice many variations of the throughput's amplitude. Therefore, when solving the optimal control problems P_1, P_2, P_3, P_4 the following Matlab functions have been used: *patternsearch* and *fmincon* with multiple initial points, and *ga* incorporated in the Genetic Algorithm and Direct Search Toolbox with multiple runs. Table 2 lists the throughput obtained when solving P_3 by using the Matlab functions *fmincon* and *patternsearch* with three initial points and when running the Matlab function *ga* three times. By comparing the throughput attained for each of the three optimization routines and the corresponding computation time, one may note that the *patternsearch* function gives the best results. Consequently, this optimization technique will be further used for solving the optimization problems.

The improvement of the throughput obtained when using optimal control with a constant speed is defined with respect to the throughput obtained when the bottom system of the flats sorting machine is static. Simulations indicate that for a set-up with one and two feeders the improvement is about 1% and 20% respectively. Hence, since the improvement obtained by using the proposed

Table 3
Comparison of throughput obtained by using the proposed control methods.

Scenario	Opt. ctrl. with pwc. speed	Opt. ctrl. with constant speed	MPC ($T_s = 5\,\mathrm{s}$ $N_c = N_p$)	MPC ($T_s = 10\,\mathrm{s}$ $N_c = 1$)
1	15.78	15.68	15.73	15.71
2	15.89	15.84	15.87	15.63
3	15.91	15.88	15.83	15.73

set-up with only one feeder is not substantial, only the systems with two feeders is further considered. The results obtained when using optimal control with a constant speed are shown in Table 3.

5.3 Optimal control with a piecewise constant speed

Table 3 lists the throughput obtained when solving P_2 for $T_s = 3\,\mathrm{s}$. Simulations show that using a smaller sampling interval does not further increase the throughput. Also, if one solves P_2 for $N > 4$ control variables, and P_1 for $2N$ control variables, the resulted throughput is the same within an accuracy of 10^{-3}. However, the computation time when solving P_1 is much larger than the one required when solving P_2.

5.4 Model predictive control

When applying MPC, the smaller T_s is chosen the bigger N_c has to be set in order to maximize the performance. If one does not want to increase N_c, blocking [3] can be also used. First improving the performance is considered and afterwards, the computational effort is taken into account.

To obtain the best throughput a maximal prediction horizon is selected, while N_c is set equal to N_p. To this aim, the time T needed to sort the entire stream of flats using optimal control with a constant speed is computed. Accordingly, the prediction horizon is set to $\left\lceil \frac{T}{T_s} \right\rceil$, while $N_c = N_p$. Various lengths T_s of the time period have been considered. Based on simulation results, it has been noticed that for $T_s \leq 5\,\mathrm{s}$ the resulting values of the throughput remain the same within an accuracy of 10^{-3}. This choice gives high performance, but is not feasible due to the computational effort. Simulations indicate that

[3] Instead of making the input to be constant beyond the control horizon only, one can force the input to remain constant during some predefined intervals.

Table 4
Comparison of the average throughput and the average computation time of the proposed control methods for a flats sorting machine with two feeders.

Control method	CPU time (s)	relative performance (%)
opt. ctrl. with pwc. speed on time intervals of variable length	$2.7 \cdot 10^7$	100
opt. ctrl. with pwc. speed on time intervals of constant length	$1.4 \cdot 10^6$	100
opt. ctrl. with constant speed	$3.5 \cdot 10^3$	99.60
MPC with $N_\mathrm{c} = N_\mathrm{p}$ and $T_\mathrm{s} = 5$s	$1.5 \cdot 10^5$	99.68
MPC with $N_\mathrm{c} = 1$ and $T_\mathrm{s} = 10$s	$2.1 \cdot 10^3$	98.93
static bottom system	0	82.80

applying MPC with a prediction horizon determined by a buffer of 120 flats, $N_\mathrm{c} = N_\mathrm{p}$, and $T_\mathrm{s} = 5$ s gives already the throughput within 1% deviation of the throughput achieved when applying optimal control with a piecewise constant speed (see Table 3). Nevertheless, high computation time is still required. Therefore, also $N_\mathrm{c} = 1$ has been considered. This choice may produce real-time, but suboptimal results.

5.5 Discussion

Based on simulations [4], a summary of the obtained results is presented in Table 4. It has been assumed that the maximal achievable throughput is obtained by using the optimal control with piecewise constant speed on time intervals of variable length. The performance of the other approaches was computed relative to this maximum. But, for each of the control methods to be compared, the throughput corresponding to the chosen scenarios varies. Therefore, the average throughput, $J = \dfrac{J_{\mathrm{scenario}\,1} + J_{\mathrm{scenario}\,2} + J_{\mathrm{scenario}\,3}}{3}$, is used in calculating the relative performance. The computation time is also averaged over all considered scenarios.

The simulation results show that applying MPC gives a good trade-off between the computation time and the maximal achievable throughput. Also, the optimal control approach is not feasible, in the sense that in reality the entire stream of flats that enter the system in one sorting round is not known in advance. Only a finite buffer b of codes is known beforehand, with b depending on the maximal time allowed to prepare the flats for sorting. Therefore,

[4] The simulations were performed on a 3.0 GHz P_4 with 1 GB RAM.

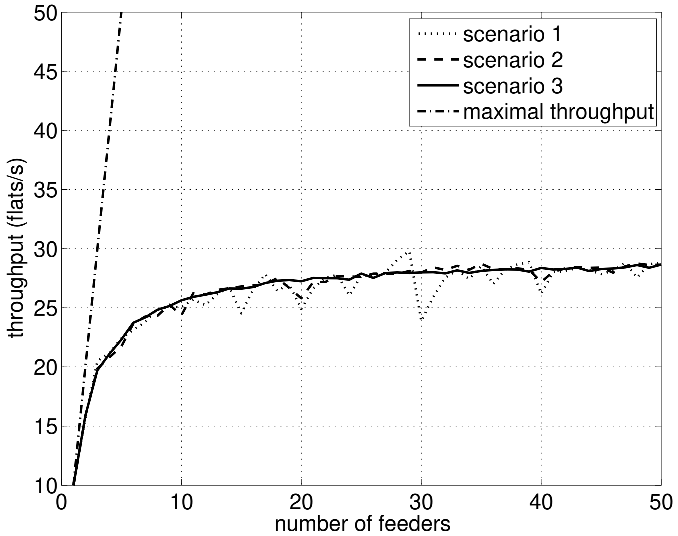

Fig. 5. Throughput versus number of feeders.

MPC is suitable in determining the optimal velocity sequence of the bottom system.

6 Influence of the structural changes

This section analyzes how the number of feeders, their position, and the velocity of the top system influence the throughput of the flats sorting machine.

6.1 Influence of the number of the feeders

Consider a flats sorting machine with F inserting devices positioned symmetrically, with the velocity of the top system being constant, equal to $1\,\mathrm{m/s}$, and the velocity of the bottom system determined using optimal control with a constant speed. The maximal throughput will be achieved for scenarios where the flats are ordered such that once they enter the system, the time that they spend in the box is negligible. In this case, the smallest possible value for the total time needed for sorting a buffer of l flats is equal to $t_{\text{sorting}} = \dfrac{(l-1)\cdot w_{\text{box}}}{v_{\text{top}}}$ where w_{box} is the width of a box.

Accordingly, for $F \cdot l = f$, the throughput is given by $J_{\max} = F \cdot \dfrac{l}{t_{\text{sorting}}}$. So, in this case ($w_{\text{box}} = 0.1\,\mathrm{m}$, $v_{\text{top}} = 1\,\mathrm{m/s}$, and f large), the maximal throughput possible is $J_{\max} = F \cdot 10\,\text{flats/s}$.

Figure 5 illustrates the maximal throughput achieved by using optimal control

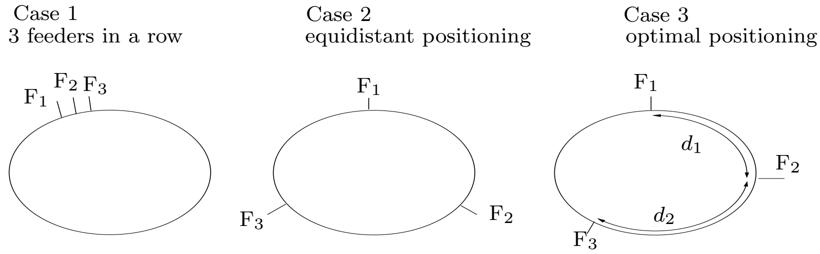

Fig. 6. Case studies.

with a constant speed to determine the optimal velocity of the bottom system when dealing with ordered, random, or combinations of random and presorted mail versus the number of feeders. Note that the throughput first increases with the number of feeders, but levels off around twenty feeders. Also, adding more feeders makes the system more complex and more expensive.

6.2 Influence of the position of the feeders

In order to analyze which is the optimal position of the $F-1>0$ feeders when the position of the first feeder is fixed, one may consider the problem of optimizing also the sequence of distances $\mathbf{d} = [d_1\ d_2\ \ldots\ d_{F-1}]^{\mathrm{T}}$, where d_j, $j = 1, \ldots, F-1$, is the distance between the jth and the $(j+1)$st inserting device. Then the optimization problem can be formulated as follows:

$$\mathrm{P}_5\colon\quad \max_{\mathbf{d}}\ J(\mathbf{d}, u^*_{\mathrm{ct}})$$

$$\text{subject to } \mathbf{t} = \mathcal{M}(\mathbf{v}, \boldsymbol{\delta}) \text{ and } \mathcal{C}(\mathbf{v}, \boldsymbol{\delta}, \delta_{\mathrm{x}}) \leq 0$$

where u^*_{ct} is the optimal constant velocity of the bottom system for configuration $\mathbf{d}$ and for the given scenario, and $\mathbf{d}$ is the sequence of distances between the feeders defined above.

Three case studies are considered, as sketched in Figure 6, where the three existing feeders of the flats sorting machine are positioned one next to the other (case 1), in an equidistant way (case 2), and at distances d_1 and d_2 computed by solving P_5 for the optimal constant speed of the bottom system for that specific scenario (case 3). The throughput obtained for each of these configurations is illustrated in Table 5. The simulation results indicate that in the first case the feeders perform like there is only one feeder, whereas by positioning them in an equidistant way one obtains the throughput within 1 % of the one obtained by solving P_5.

Table 5
J_{opt} (flats/s) obtained when positioning the feeders according to the considered case studies.

Scenario	Case 1	Case 2	Case 3
1	9.98	20.49	20.49
2	9.97	19.87	19.94
3	9.94	19.71	19.85

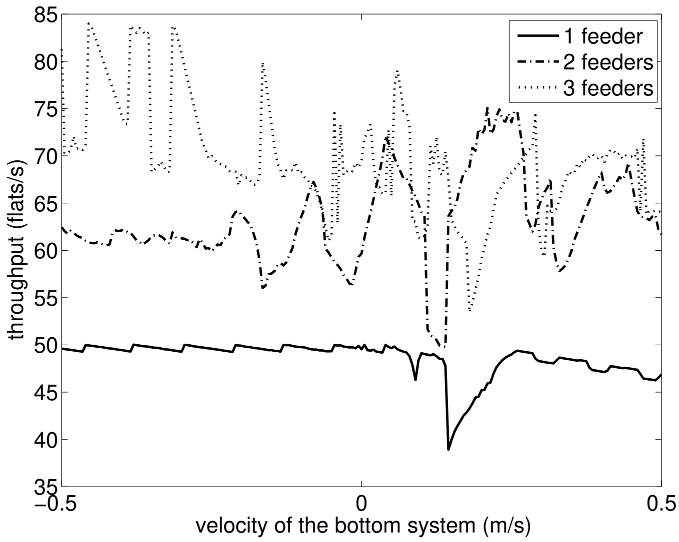

Fig. 7. Throughput versus v_{bottom} when $v_{\text{top}} = 5\,\text{m/s}$.

6.3 *Influence of the velocity of the top system*

One may consider that the inserting process can be performed even for a maximal velocity $v_{\text{top}} = 5\,\text{m/s}$. In Figure 7, the throughput of the flats sorting machine is plotted versus the velocity of the bottom system. The velocity had been discretized with the sampling step of $0.005\,\text{m/s}$. Note that only the curves for the second scenario have been plotted, since the plots for the first and third scenario are similar. One may notice that only increasing the relative speed between the top and the bottom system does not maximize the thoughput. As a consequence, implementing advanced control methods to compute the optimal velocity of the bottom system is still required.

7 Conclusions and future work

This paper has started with a short description of how flats sorting machines currently work. Afterwards, a new set-up has been proposed by making minor design changes i.e. adding extra feeders and moving the bottom bin system.

An event-driven model of the process has been determined, and advanced control methods have been implemented so as to ensure the optimal speed of the movements.

It has been analyzed how the number of feeders, their position, and the velocity of the top system influence the throughput of the automated flats sorting machine. The simulation results show that just increasing the speed of the top system, and hence, the relative speed between the top and bottom system does not have as immediate consequence an increase in the throughput. Hence, determining the optimal bottom velocity is still required so as to maximize the efficiency of the flats sorting machine. The results indicate that model predictive control is the most appropriate control method to determine the velocity of the bottom system for the proposed flats sorting machine. However, the performance is then influenced by the prediction horizon and the control horizon.

In future work also other (receding horizon) control methods will be considered such as fast heuristic approaches, fuzzy control, case-based control, etc. Also more complex dynamics of the system than those considered in this paper will be included.

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

Audet, C., Denis Jr., J.E., 2002. Analysis of generalized pattern searches. SIAM Journal on Optimization 13 (3), 889—-903.

Chiang, J., Gader, P., 1997. Hybrid fuzzy-neural systems in handwritten word recognition. IEEE Transactions on Fuzzy Systems 5 (4), 497–510.

Gill, P., Murray, W., Wright, M., 1981. Practical Optimization. Academic Press, London, UK.

Goldberg, D., 1989. Genetic Algorithms in Search, Optimization, and Machine Learning. Addison-Wesley, Boston, Massachusetts, USA.

Kaneko, H., Fujiwara, E., 2004. A class of m-ary asymmetric symbol error correcting codes for data entry devices. IEEE Transactions on Computers 53 (2), 159–167.

Leimer, J., 1962. Design factors in the development of an optical character recognition machine. IEEE Transactions on Information Theory 8 (2), 167–171.

Lewis, F., 1986. Optimal Control. John Wiley & Sons, New York, USA.

Lohmann, B., 1996. Throughput control for a transport process and an application in postal automation machines. Control Engineering Practice 4 (11), 1503–1509.

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

Whichello, A., Yan, H., 1996. Locating address blocks and postcodes in mailpiece images. In: Proceedings of the 13th International Conference on Pattern Recognition. Vol. 3. pp. 716–720.