Delft University of Technology

Delft Center for Systems and Control

Technical report 06-037

Multi-agent controllers for large-scale transportation systems – Application to postal automation*

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

If you want to cite this report, please use the following reference instead:
A. Tarău, B. De Schutter, and H. Hellendoorn, "Multi-agent controllers for large-scale transportation systems – Application to postal automation," *Proceedings of the 9th TRAIL Congress 2006 – TRAIL in Motion – CD-ROM*, Rotterdam, The Netherlands, 12 pp., Nov. 2006.

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

Multi-Agent Controllers for Large-Scale Transportation Systems

Application to postal automation

TRAIL Research School, Delft, November 2006

Authors

M.Sc. Alina Tarău, dr.ir. Bart De Schutter and prof.dr.ir. Hans Hellendoorn

Faculty of Mechanical, Maritime and Materials Engineering, Delft Center for Systems and Control, Delft University of Technology

Contents

Abstract

During the last decades, transportation systems have known fast growing volumes of transporting materials and the focus is on quality, reliability and throughput maximization. The throughput of these automated sorting and transporting machines is limited by mechanical capabilities and also by the performance of the process devices (surface scanners, address reading devices, bar-code reading devices, etc.). What we require are structured distributed control strategies, applicable for general transportation systems, characterized by materials being processed, while they are transported by conveyor systems e.g. sorting machines, baggage handling, distribution systems. The goal is e.g. to maximize the throughput. In this paper we take postal automation as an example of transportation system. We give an overview on how postal automation works, we define the control problems and list some important open problems in controlling the system. Finally, we propose possible approaches to address some of these open problems.

Keywords

transportation systems, distributed control, postal automation, throughput control

1 Introduction

In the last decades, there has been a growing interest in modeling and controlling large-scale hybrid systems. A hybrid system is a dynamic system which exhibits both continuous and discrete dynamics. If we consider, e.g. a conveyor system then the transport of parcels on the conveyors can be modeled as a continuous process, characterized by e.g. the speed of the conveyor, which can in principle be adjusted continuously. Actions like feeding a parcel on the belt, removing the parcel, rerouting it, etc. provide discrete actions on the system. What we are interested in, is how to control these large-scale hybrid systems. Up to now, most control methods for hybrid systems are based on centralized control and/or on ad-hoc techniques. However, centralized control of large-scale systems is often not feasible in practice due to computational complexity, communication overhead, and lack of scalability. A structured control design method is also lacking. Therefore, our aim is to develop a structured and tractable design methodology for robust control of large-scale hybrid systems.

More specifically, we refer at transportation systems like sorting machines, baggage handling, and distribution systems. They can be modeled as a hybrid system. Typical issues of this application are: coordination of the processing units, task allocation, time scheduling, prevention of jams and deadlocks, buffer overflows, maximization of the throughput, avoid damage of the goods, cost minimization, etc. The issues which we proposed to investigate are typical for a general class of transport processes, characterized by materials being processed, while they are transported by conveyor systems. In this paper we will focus on postal automation, as an example of transportation system and later on we will extend the problem of controlling the postal sorting system to baggage handling or other sorting systems.

We require a control strategy that can deal with the coupling between tasks. Therefore distributed control is needed. Several techniques have already been developed, see e.g. Siljak (1991), Fusrikov (2000), Kosakaya & Yamaoka (2000), Mutambara (1998), Popovic & Bhatkar (1990), etc.

Furthermore, we propose to incorporate a hierarchical multi-agent framework. Hierarchical control is a compromise between centralized and decentralized control, combining the advantages of both (global system performances and tractability). In multi-agent systems (Weiss (1999), Russell & Norvig (2003), Sycara (1998), Vlassis (2003), Kosakaya & Yamaoka (2000), Russel & Subramanian (1995)) the control is divided among autonomous agents, each controlling and monitoring an individual unit, or a group of neighboring units, being able to communicate and to cooperate. Therefore the systems chosen to be modeled and controlled will be split in subsystems, each having a local controller. Those will be supervised and controlled by a higher-level controller or supervisor.

We may also use Model Predictive Control (MPC), see e.g. Qin & Badgwell (2002), Allgower et al. (1999), Camacho & Bordons (1995) which can be combined with multi-agent control (Negenborn et al. (2006a), Negenborn et al. (2006b)), or distributed control (Camponogara et al. (2002)). According to Maciejowski (2002) MPC is a popular control methodology in the process industry, providing many attractive features: it is an easy-to-tune method, it can handle constraints in a systematic way, it is applicable to multi-variable systems, and is capable of tracking pre-scheduled reference signals. In

MPC at each sample step the optimal control inputs that minimize a given performance criterion over a given prediction horizon are computed, and applied using a receding horizon approach.

This paper is organized in the following way. In Section 2, we present a survey[1] on postal automation, describing the state of the art. Section 3 deals with general problems of postal sorting machines. In Section 4 we present possible directions for solving the problems either by making minor design changes or by using a control based approach. Here we introduce and formalize the terms of model predictive control and multi-agent. Finally, in Section 5 we conclude and present the future work.

2 Postal automation

2.1 Manual mail/flats processing

There are two main categories of letters: mail and flats. By mail we understand regular letters, while flats are large letters (A4 size envelopes) with more or less information printed on it (sender and destination address, advertisement messages, stamps and mail class service information), plastic-wrapped mail items, magazines, newspapers or catalogs. Moreover in the flat category are often included also small parcels with a maximum thickness of about 40 mm.

Manual mail/flats processing involves series of operations, with human hands at work every step of the way.

As it can be seen in Figure 1, the process begins with posting mail/flats in post boxes. The postmen collect the mail items and transport them to the post office. There the culling process starts. This means that the mailed items are separated into different streams like letters, flats and packets. The letters/flats are faced in the same way, and in the canceling section the postal stamps used for postage are voided so that the stamp cannot be used again. In the sorting phase, the mail/flats items are sorted (first by category: express, registered, etc. and later on based on postal code and/or address). Afterward, in the carrier sequencing phase, they are deposited in the corresponding bins. Finally, they are delivered to the customers.

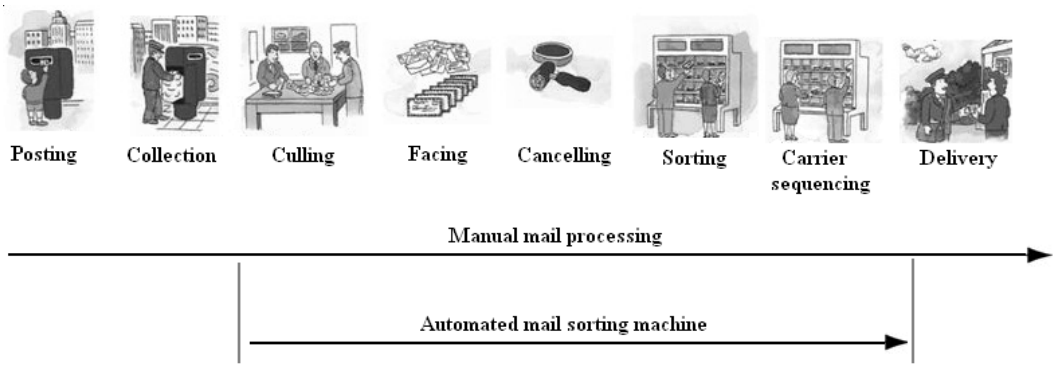

Figure 1: Manual versus automated mail processing

[1]This survey was performed as the first step of the first author's PhD project.

2.2 Automated mail processing

As illustrated in Figure 1, automated mail processing can handle many operations in an automated way (starting with culling the mail and finishing with carrier sequencing) leading to better efficiency.

Before explaining how the automated mail sorting process works, we will notice that it can be divided in two parts: first, mail preparation, then coding and sorting.

Mail preparation involves the following steps:

- culling, to separate out the items that cannot be handled by automated systems (what is not regular letter).
- facing, to ensure that mail items are faced and oriented accordingly, simplifying subsequent processing.
- canceling.
- sorting by category (express, registered, etc.).

When mail leaves the mail preparation process, it moves into the coding/sorting process. The sorting information is printed onto the letter in form of a bar-code. Finally, based on the bar-code, the mail will be allocated to specific postal delivery routes.

Every letter is automatically read, whether its address is machine printed or handwritten. The high-performance reading technology used is Optical Character Recognition (OCR). The procedure is the following: OCR tries to read all the information from the addresses or bar-codes. The information is printed in a bar-code form. If there is no machine-readable information, an image of the letter will be transmitted automatically to the video coding system. Operators view the address image on a monitor, read the delivery information and enter it via a keyboard. A transport delay line of several seconds (which can be increased) allows the system to deliver sorting information on-line before the mail item reaches the code printing and outlet sections.

If a longer time is needed to read the address since the image is processed off-line, an identification code will be assigned to the mail and the letter will be sent to a special stacker until the image has been processed. Therefore an auxiliary feeder can be attached to balance the combined on-line OCR and video coding system process, increasing the efficiency.

2.3 Automated flats processing

A flats sorting system performs all mail sorting functions for large letters, plastic-wrapped mail items, magazines and catalogs. The feeder device only receives flats and therefore the process is similar to that for regular letters, except for the culling phase, which does not exist in here.

The main difference between the two automated sorting systems is the sorting part. In the regular letter case, the mail is transported through a vertical conveyor system to the specific destination. In contrast to this way of sorting, the flat piece is inserted into a transport box (cassette) of the sorting section by the inserting device. The transport box carries flat pieces and sorts them into destination trays according to the selected sorting scheme. Finally, when the tray becomes full, it is replaced by an empty one.

3 Problems

3.1 Control problems

The throughput of a mail piece sorting machine is limited by the mechanical capabilities of the machine and also by the performance of the address reading device. Therefore, one of the most important problems to be controlled for our sorting system is the feeding rate. The feeding rate (in quantity/second) is a control input. This has to be set as high as possible (in order to maximize the throughput), and low enough in order to avoid unprocessed mail items reaching the end of the delay line (buffer overflow). One can notice that once a mail item has been fed, it cannot be stopped since the conveyor is moving with constant speed.

In Lohmann (1996), this problem has been controlled using a distributed parameter system with boundary and initial condition as a mathematical model.

The problem of locating the block of typed or handwritten destination address and extracting the necessary information has been treated to a very large extent (see e.g. Whichello & Yan (1996), Gader & Khabou (1996), Koerich et al. (2005), L. Heutte & Hernoux (2000))

To the authors' best knowledge, there are no papers that describe how to solve the route assignment problem for postal automation applications, but it is solved (inside the company), the conveyor system having constant speed.

3.2 Open problems

At the moment, some of the typical issues in a postal center are the following:

- there is too much damaged mail in the culling phase.
- jams appear in the sorting phase (which can only be solved just by stopping the machine, eliminating the jammed mail item (reroute it) and starting the system again).
- a lot of operators are needed at the video coding desk (a solution being to improve the OCR or to increase the deadline with the cost of slower processing).
- many operators are also needed for emptying the bins into trays, in the storing mail section.

4 Possibilities for improvement

4.1 A new set-up

In order to solve the open problems of postal automation we can make minor design changes.

To prevent damage of the mail a new automation for sorting mails to categories can be designed. In this section we will propose such a set-up. One can use a conveyor system for transporting the mail and sensors for checking the width, length, height and weight of each of the items. When there is one which does not correspond to a regular letter, it will be pushed to another conveyor so as to collect them in the non-machinable tray.

All the automation discussed in Section 2.2 can be put together, obtaining a compact letter sorter. This system is based on the design of a compact reader sorter with facing, canceling, reading and coding functions, linked together with the carrier sequence sorter[2].

In order to use less operators for emptying the bins, a new conveyor system can be designed, such that full trays will be automatically replaced with empty ones. The full tray will be pushed in front by a simple pushing system. The new conveyor system will be placed behind the one where the trays used for depositing the sorted mail items are, as it is illustrated in Figure 2. We propose a multi-actuator conveyor, because we want to move the trays one by one, if there is an empty space on the conveyor as a result of replacing a full tray.

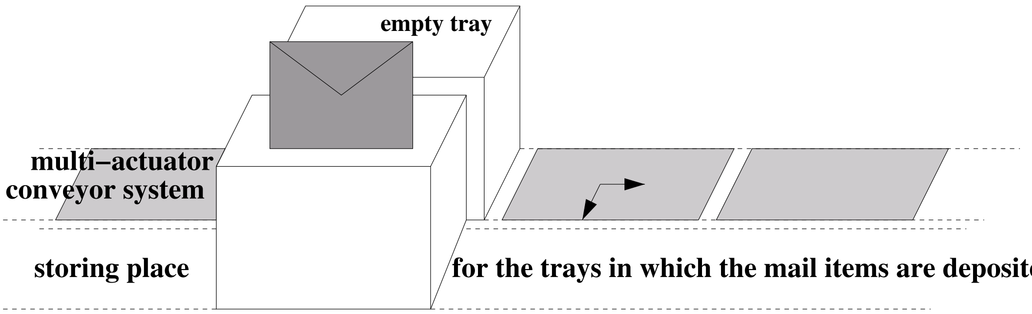

Figure 2: Replacing full trays

This multi-actuator conveyor system can move trays to the right (to assure empty trays for all destinations at all time instants) or in front (to replace a tray when it becomes full). For detecting when a tray becomes full, we propose to use sensors attached to each destination route, detecting the fill level. We consider a tray to be full if the height of the stack of flats in the tray reaches e.g. 95% of the actual height of the tray. The tray will be moved to the right if there is an empty place and the tray in front has more than a certain free storage (we can e.g. assume more than 5% free storage). An algorithm has to be implemented to move the trays on the multi-actuator conveyor system avoiding deadlocks. In this case, a deadlock might appear when there is no more mail for the tray which has less than 5% free capacity (and therefore the tray cannot move to the right) but tray is needed on the right side. This can be avoided if we can predict the model of incoming mail (arrival hours, number of consequently mail items) for each destination.

Regarding the flats sorting machine, we can design a new system. The transport box system will move as usual, with constant speed, but the tray system, will move to the left, to the right or not move at all, in order to maximize the number of empty transport cassettes. The transport system is illustrated in Figure 3. For maximizing the throughput a second feeder device will be used. This is useful when the machine is fed with presorted mail.

[2]The carrier sequence sorter is the automation that deals with storing the mail into the corresponding bin.

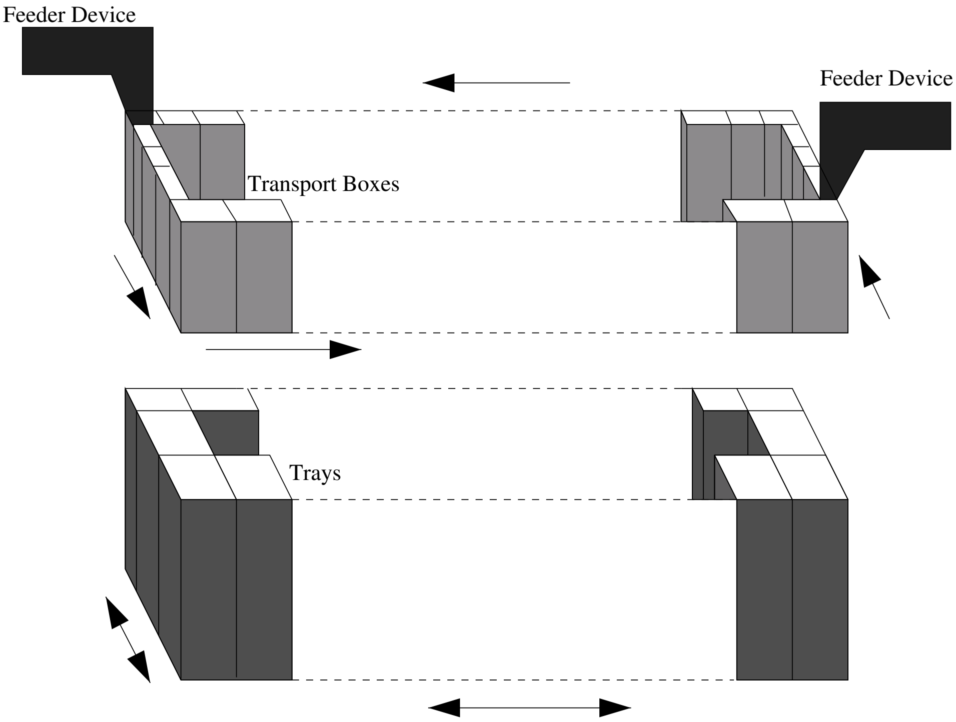

Figure 3: Sorting part of a flats sorting machine

4.2 Control based approach

4.2.1 Model predictive control

In the previous section we have seen that we can solve some open problems by making minor design changes. In this section, we propose control approaches to solve some control problems for the existing set-up.

We introduce the control methodology called *Model Predictive Control* (MPC) see e.g. Qin & Badgwell (2002), Allgower et al. (1999), Camacho & Bordons (1995), Maciejowski (2002). MPC can be used for setting the feeding rate of the sorting machine. As for mail it is not easy to predict if the address and/or postal code are easily read by OCR or not, we can use this technique for flat sorting machine.

In order to solve the deadlock discussed in Section 4.1, we only need prediction of the incoming mail model (for each destination), because what we want to optimize are the trays' movement on the multi-actuator conveyor system. The goal is to assure an empty tray (as back-up) for each destination.

We will now explain MPC in more detail, the working principle being illustrated in Figure 4.

MPC is an on-line controller design method, in which on step k, the model is used to predict the future behavior of the system over a given prediction horizon $[k, k + N_\mathrm{p}]$ for a given input sequence and where a cost criterion J is optimized subject to constraints on the inputs and outputs.

Predictive control uses a receding horizon principle, where the control input is obtained by solving a discrete-time optimal problem over a control input sequence. This means that after computation of the optimal control sequence, only the first control sample will be implemented, subsequently the horizon is shifted. So, the model is regularly

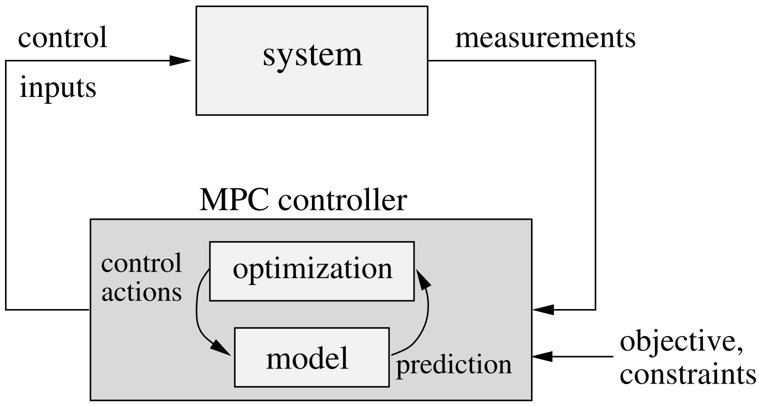

Figure 4: MPC - working principle

updated based on the measurements coming from the sensors and the control inputs are computed using this new information. The idea of receding horizon is illustrated in Figure 5. At time k the future control sequence $u(k|k), ..., u(k+N_c-1|k)$ is optimized such that the performance index J is minimized subject to the constraints. At time k the first element of the optimal sequence $(u(k) = u(k|k))$ is applied to the real process. At the next time instant the horizon is shifted and a new optimization at time $k+1$ is solved.

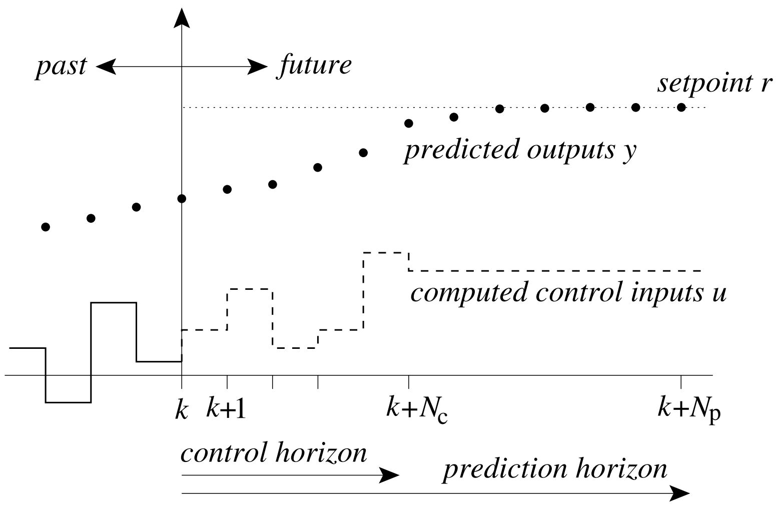

Figure 5: Conventional MPC. At time k the future control sequence $u(k|k), ..., u(k+N_c-1|k)$ is optimized such that the performance index J is minimized subject to the constraints. Here the performance J to be minimized is the difference between the output and the set points.

The control problem described in Figure 5 is to find actions $u(k)$,...,$u(k+N_c)$, such that after N_p steps the system behavior y approaches the desired behavior y^*.

In this research we will develop an approach to design controllers for multi-actuator continuous transport systems. The main goal of the controller is to obtain coordination and synchronization between the processing units, so as to maximize the throughput. Additionally constraints will arise in order to avoid unprocessed items, damage of the goods, etc.

For setting the feeding rate we will use is a single-agent MPC. The problem can be formulated in the following way:

find the feeding rate that maximize the throughput

subject to

the model of the system
the time [3] *that OCR needs for reading the address and/or postal code*
the bounds[4] *on the feeding rate.*

Another way to see the second constraint is the following: the flat travel distance from the scanner to the current position should be smaller than the total travel distance.

The main advantage of MPC is that it is an easy-to-tune model-based control approach that can effectively deal with constraints on the inputs, outputs and states of the system. The actual control objectives and operating constraints can be represented explicitly in the optimization problem which is solved at each control instant.

4.2.2 Multi-agent control

In order to control a stand alone sorting machine, centralized control can be used. However, when considering large-scale sorting systems, we will deal e.g. with a large number of coupled postal sorting machines, which may have more than one feeder device. Due to computation complexity, necessity of communication and scalability, we propose to use a multi-agent approach, see e.g. Weiss (1999), Russell & Norvig (2003), Sycara (1998), Vlassis (2003), Kosakaya & Yamaoka (2000), Russel & Subramanian (1995) and we will combine it with model based predictive control.

Accordingly to Russell & Norvig (2003), an *agent* is any entity that can perceive its environment through sensors and act upon it through actuators, having goals and limited knowledge. This definition includes humans agents who perceive the living environment through their senses and act upon it using their body; robotic agents, which have cameras as sensors and wheels; motors or grippers as actuators; software agents, which receive entry data and act corresponding to the implemented program, and so on. The term of autonomous will be used to refer to an agent capable of making independent decisions and taking actions to satisfy internal goals based mainly on its own perception rather than on prior knowledge given to it at design moment.

A system composed by a group of agents which can eventually interact with each other is called *Multi-Agent System* (MAS). In a MAS, each agent in general has incomplete information, being restricted to its capabilities, data is decentralized, system control is distributed, while the computation is asynchronous (parallel or serial).

In this section, we consider an episodic environment, according to the application which we have chosen to model and to control. In an episodic environment, the state history generated by the actions of the agent can be considered as divided in episodes, each of which is terminated by an action. This kind of environment is simpler, in the sense that each agent decides what action to perform only based on the current episode. Each

[3] There is a maximal time allotted to the OCR to read the address, which is $\frac{d}{v}$, where d is the travel mail/flat distance from the OCR scanner to the sorting section, and v is the constant speed of the conveyor.

[4] The feeding rate is bounded by 0 and a maximal feeding rate, which will be computed taking into account the mechanical capabilities of the machine.

episode starts with the arrival of a new mail/flats, and ends with the execution of the physical action recommended by the sorter: routing the piece to the specific tray.

Definition 1 *An agent is, in this case, a tuple* $\langle X, O, A, f, f_{\mathrm{p}}, f_{\mathrm{a}}, x_0 \rangle$*, where:*

- X *is the internal state space of the agent.*
- O *is the observations (perceptions) space of the agent.*
- A *is the agent's action space.*
- $f : X \times O \rightarrow A$ *is the agent's function, which is mapping from perception sequences to actions.*
- $f_{\mathrm{p}} : S \rightarrow O$ *is the perceptual function of the agent, which is a mapping from the environment state to the agent's perceptions, where* S *is the environment's state space.*
- $f_{\mathrm{a}} : X \times O \rightarrow X$ *is the agent's transition function, which is describing how the agent evolves as a result of the environment's perception.*
- x_0 *is the initial state of the agent.*

As we already noticed, an agent function maps perceptions of the environment states into actions. If we see it as a controller, the perceptions represent the input or the control feedback and the actions are command output.

Before defining the multi-agent system, which contains a set of agents, we introduce some additional notations. Let Λ be the set of MAS agents. An agent in the set Λ will be denoted indexed by i. We will only consider the discrete time case, therefore, the current value of the discrete time variable will be denoted by k. To refer to the ith agent's observation that it can perceive at any instant k, we will use the conventional notation $o_i(k) \in O_i(k)$, $a_i(k) \in A_i(k)$ is the agent's action, $x_i(k) \in X_i(k)$ the internal state and $s(k) \in S(k)$ the environment's state, all at step k. The joint actions of all the agents $i \in \Lambda$ will be denoted by $a^{\text{joint}} = [a_1, a_2 ... a_{\dim \Lambda}]^{\mathrm{T}} \in A^{\text{joint}} = A_1 \times A_2 ... \times A_{\dim \Lambda}$.

Definition 2 *A MAS is a tuple* $\langle \Lambda, S, f_{\mathrm{e}}, s_0 \rangle$*, where:*

- Λ *is the set of agents.*
- S *is the environment's state space.*
- $f_{\mathrm{e}} : S \times A^{\text{joint}} \rightarrow S$ *is the environment's transition function, which is describing how the environment evolves as a result of the agents' actions.*
- s_0 *is the initial state of the environment.*

The environment is perceived through the function f_{p}. This determines the agent to update its internal state $X_i(k)$ with respect to the perception of the environment as soon as it can be observed to $X_i(k+1) = f_{\mathrm{a}}(X_i(k), O_i(k))$. The updated internal state is used for determining the action choice. The environment will change as a result of the agents' actions upon it and the cycle will start again. We will write an abstract algorithm to describe this interaction between the agents and their environment.

Algorithm 3.1 MAS evolution:

$k \leftarrow 0$
loop
$o_i(k) = f_\mathrm{p}(s(k))$ perceive with respect to environment
$x_i(k+1) = f_\mathrm{a}(x_i(k), o_i(k))$ the agent's state evolves
$a_i(k) = f(x_i(k+1), o_i(k))$ action is taken
$s(k+1) = f_\mathrm{e}(s(k), a^\mathrm{joint}(k+1))$ the environment's state evolves
$k \leftarrow k+1$
end loop

Another important issue to discuss is the *communication* between the agents. We will add a communication channel to the environment, through which agents will receive and/or send messages to the environment.

The only differences in the agent's definition will be the following:

- the perception space of the agent will be $O = O^\mathrm{e} \times M^\mathrm{rcv}$, where O^e will represent the space of environment observations and M^rcv will be the received message space of the agent (a state of the observation space being: $o = [(o^\mathrm{e})^\mathrm{T}, (m^\mathrm{rcv})^\mathrm{T}]^\mathrm{T}$).
- the actions space of the agent will be $A = A^\mathrm{e} \times M^\mathrm{snd}$, where A^e will represent the space of actions of the agent on the environment, while M^snd will be the sent message space of the agent (a state of the action space being: $a = [(a^\mathrm{e})^\mathrm{T}, (m^\mathrm{snd})^\mathrm{T}]^\mathrm{T}$).

Definition 3 *A communicative MAS is a multi-agent system, where:*

- Λ *is the set of agents.*
- $S = S^\mathrm{e} \times S^\mathrm{c}$*, where* S^e *is the state space of the environment, and* S^c *is the state space of the communication channel (a state space of the environment with communication channel being:* $s = [(s^\mathrm{e})^\mathrm{T}, (s^\mathrm{c})^\mathrm{T}]^\mathrm{T}$*).*
- $f_\mathrm{e} : S \times A^\mathrm{joint} \to S$ *is the transition function, which describes how the environment evolves as a result of the agents' actions and communication. The joint actions of all the agents* $i \in \Lambda$ *will be denoted by*
  $a^\mathrm{joint} = [[(a_1^\mathrm{e})^\mathrm{T}, (m_1^\mathrm{snd})^\mathrm{T}]^\mathrm{T}, [(a_2^\mathrm{e})^\mathrm{T}, (m_2^\mathrm{snd})^\mathrm{T}]^\mathrm{T} ... [(a_{\dim\Lambda}^\mathrm{e})^\mathrm{T}, (m_{\dim\Lambda}^\mathrm{snd})^\mathrm{T}]^\mathrm{T}]^\mathrm{T} \in A^\mathrm{joint}$
- s_0 *is the initial state of the environment with communication channel.*

Therefore, assuming this communication channel embedded in the environment, its states (s^e) and messages (m_i^rcv) are read by the agent i, the state of the agent is updated, decision actions are taken and messages are sent to the environment. The action result will be updating the environment state, while the rest of the agents will receive the broadcast message.

The evolution of the communicative MAS follows the same lines as Algorithm 3.1 (with appropriate replacement of the notations).

In the definitions and algorithms above we have considered an accessible environment (one in which the agents can obtain complete, accurate, up-to-date information about the environment's state).

5 Conclusions and future work

In this paper we have given an overview of how postal automation works, what the control and open problems are, and also some techniques that could be used. This has led to the identification of certain problems and attributes at a rather non-mathematical level.

We have presented few techniques which will be used to model and control some of the typical problems of postal sorting machines. We have introduced and formalized the terms of multi-agent and model predictive control. Multi-agent control was combined with model predictive control for setting the machines' feeding rate.

As we have already mentioned, we are interested in a general class of transportation systems, characterized by materials being processed, while they are transported by conveyor systems e.g. sorting machines, baggage handling, distribution systems. Therefore, in the future we will write a general model of the transportation systems, apply the discussed approaches and analyze the system, make simulations for it and compare them with the results obtained in practice.

Acknowledgments

This research is supported by the VIDI project "Multi-Agent Control of Large-Scale Hybrid Systems" (DWV.6188) of the Dutch Technology Foundation STW, Applied Science division of NWO and the Technology Programme of the Dutch Ministry of Economic Affairs, by the BSIK project "Next Generation Infrastructures (NGI)", and by the Transport Research Centre Delft.

References

Allgower, F., T. Badgwell, S. Qin, J. Rawlings, S. Wright (1999) Nonlinear predictive control and moving horizon estimation - an introductory overview, in: *Advances in Control: Highlights of ECC'99*, London, UK: Springer, pp. 391–449.

Camacho, E., C. Bordons (1995) *Model Predictive Control in the Process Industry*, Springer-Verlag, Berlin, Germany.

Camponogara, E., D. Jia, B. H. Krogh, S. Talukdar (2002) Distributed model predictive control, *IEEE Control Systems Magazine*, 22(1), pp. 44–52.

Fusrikov, A. V. (2000) *Optimal Control of Distributed Systems : Theory and Applications*, Boston, US: American Mathematical Society.

Gader, P. D., M. A. Khabou (1996) Automatic feature generation for handwritten digit recognition, *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 18(12), pp. 1256–1261.

Koerich, A. L., R. Sabourin, C. Y. Suen (2005) Recognition and verification of unconstrained handwritten words, *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 12(10), pp. 1509–1522.

Kosakaya, J., K. Yamaoka (2000) Cooperative multi-agent intelligent field terminals, for distributed control systems, *IEICE Transactions on fundamental of Electronics, Comunications and Computer Sciences*, E83-A(11), pp. 2264–2277.

L. Heutte, A. N., T. Paquet, C. Hernoux (2000) Handwritten text recognition using a multiple-agent architecture to adapt the recognition task, *Proceedings of the 7th International Workshop on Frontiers in Handwriting Recognition, Amsterdam, Netherlands*, pp. 413–422.

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

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

Mutambara, A. G. (1998) *Decentralized Estimation and Control for Multisensor Systems*, Boca Raton, Florida: CRC Press LCC.

Negenborn, R., B. De Schutter, H. Hellendoorn (2006a) Multi-agent model predictive control for transportation networks: Serial versus parallel schemes, in: *Proceedings of the 12th IFAC Symposium on Information Control Problems in Manufacturing (INCOM'2006)*, Saint-Etienne, France, pp. 339–344.

Negenborn, R., B. De Schutter, H. Hellendoorn (2006b) Multi-agent model predictive control for transportation networks with continuous and discrete elements, Tech. Rep. 06-014, Delft Center for Systems and Control, Delft University of Technology, Delft, The Netherlands, accepted for the 11th IFAC Symposium on Control in Transportation SystemsIFAC Symposium on Control in Transportation Systems (CTS 2006), Delft, The Netherlands, August 2006.

Popovic, D., V. P. Bhatkar (1990) *Distributed Computer Control for Industrial Automation*, New York, US: Marcel Dekker.

Qin, S. J., T. A. Badgwell (2002) A survey of industrial model predictive control technology, *Control Engineering Practice*, 11, pp. 733–764.

Russel, S. J., D. Subramanian (1995) Provably bounded-optimal agents, *AI Research*, 2, pp. 575–60.

Russell, S., P. Norvig (2003) *Artificial Intelligence: A Modern Approach*, 2nd ed., Englewood Cliffs, New Jersey: Prentice Hall.

Siljak, D. D. (1991) *Decentralized Control of Complex Systems*, San Diego, California: Academic Press, INC.

Sycara, K. P. (1998) Multi-agent system, *AI Magazine*, 19, pp. 79–92.

Vlassis, N. (2003) A concise introduction to multi-agent systems and distributed AI, Tech. rep., University of Amsterdam, http://staff.science.uva.nl/ṽlassis/cimasdai/.

Weiss, G. (1999) *Multiagent Systems*, London, UK: The MIT Press Cambridge.

Whichello, A. P., H. Yan (1996) Locating address blocks and postcodes in mail-piece images, in: *Proceedings of the 13th International Conference on Pattern Recognition*, pp. 716–720.