Proceedings of the
47th IEEE Conference on Decision and Control
Cancun, Mexico, Dec. 9-11, 2008

Nonlinear Predictive Control for the Management of Container Flows in Maritime Intermodal Terminals

A. Alessandri, C. Cervellera, M. Cuneo, M. Gaggero

Abstract— **The increase of efficiency in the management of container terminals is addressed via a predictive control approach to allocate the available handling resources. The predictive control action results from the minimization of a performance cost function that measures the lay times of carriers over a forward horizon. Such an approach to predictive control is based on a model of container flows inside a terminal as a system of queues. Binary variables are included into the model to represent the events of departure or stay of a carrier, thus the proposed approach requires the on-line solution of a mixed-integer nonlinear programming problem. Two techniques for solving such a problem are proposed that account for the presence of binary variables as well as nonlinearities into the model and the cost function. The first relies on the application of a standard branch-and-bound algorithm. The second is based on the idea of dealing with the decisions associated with the binary variables as step functions. In this case, real nonlinear programming techniques are used to find a solution. Finally, a third approach is proposed that is based on the idea of approximating off line the feedback control laws that result from the application of the two previous approaches. The approximation is made using a neural network, which allows one to construct an approximate feedback control law and generate the corresponding on-line control action with a small computational burden. Simulation results are reported to compare such methodologies.**

I. INTRODUCTION

The problem of an efficient management of container terminals has been the objective of intensive investigation from time. Queueing theory may be used for performance evaluation [1], but it suffers from a poor capability of describing the dynamic behavior of the container flows inside a terminal. A similar criticism concerns the approaches based on a pure static modeling (see, e.g., [2]). As a consequence, more powerful modeling paradigms were proposed that allow one to account for dynamic aspects like, for example, discrete-event systems [3]. Discrete-event tools allow one to construct very precise models of the logistic operations carried out in container terminals, but they are quite demanding from the computational point of view. Such difficulty arises particularly when a model is used to control a container terminal in real time.

In this paper, in lines of previous works (see [4], [5]), a novel approach to container terminals modeling is proposed that is based on a nonlinear dynamic model of the terminal activities, and consists in optimizing the container flows using a limited amount of resources (i.e., cranes, yard trucks, and other transfer machines). Using this model, an optimal allocation of such resources is searched that minimizes a given performance index according to a predictive-control strategy. Predictive control is based on the idea of solving an open-loop finite-horizon optimal control problem at each time step and applying only the first control action (see [6] for an overview). The optimization has to be performed over a forward horizon from the current time instant using the available information on the container occupancies in the various areas of the terminal, and the foreseen import and export flows of the various carrier classes (i.e., typically, for a maritime terminal, ships, trucks, and trains).

The contribution of this paper can be summarized as follows. First of all, a novelty is the use of nonlinear predictive control to manage the handling activities in a maritime terminal, as a generalization of the results presented in [4], [5]. The problem we deal with is a mixed-integer nonlinear programming one, for which we investigated the use of two methodologies. The first approach consists in using a standard branch-and-bound technique for nonlinear programming [7]. The second approach is based on the idea of treating the decisions associated with the binary variables that model the departure or stay of a carrier as nondifferentiable step functions. Thus, real mathematical programming tools are required to perform the optimization.

Another contribution of the paper consists in applying an approximation scheme to find suboptimal feedback control laws. Such laws result from the approximation of the optimal ones that are obtained via the two above-described methodologies. Among the various choices for the family of nonlinear approximators, we focused on neural networks (see, e.g., [8]). Such class of approximators includes one-hidden-layer neural networks, which exhibit another powerful feature that consists in requiring a small number of parameters (i.e., the neural weights) to ensure a fixed approximation accuracy, especially in high-dimensional settings. More specifically, one-hidden-layer sigmoidal neural networks and radial-basis-function networks with tunable external and internal parameters may guarantee a good uniform approximation with upper bounds depending on a number of parameters that grows at most polynomially with the dimension of the input of the function to be approximated (see [9]–[11] and the references therein). According to such an approach, we constructed suboptimal controllers that take on the structure of a neural network, whose parameters are chosen by training algorithms that allow one to approximate the optimal control laws.

A. Alessandri and M. Gaggero are with the Department of Production Engineering, Thermoenergetics, and Mathematical Models (DIPTEM), University of Genoa, P.le Kennedy Pad. D, 16129 Genoa, Italy (email: alessandri@diptem.unige.it, gaggero@diptem.unige.it). C. Cervellera and M. Cuneo are with the Institute of Intelligent Systems for Automation (ISSIA), National Research Council of Italy, Via De Marini 6, 16149 Genoa, Italy (email: cervellera@ge.issia.cnr.it, marta@ge.issia.cnr.it).

978-1-4244-3124-3/08/$25.00 ©2008 IEEE

The learning process is computationally demanding, but it is made off line. By contrast, the on-line use of the resulting neural controller requires a very small computation.

II. A DYNAMIC MODEL OF TERMINAL OPERATIONS

The model of container flows considered in this paper (see Fig. 1 for a pictorial sketch) is derived by the modeling framework reported in [5]. The arrivals of ships, trucks, and trains are modeled by means of waiting queues (q_{ship}, q_{truck}, and q_{train}), where the various carriers stay until a berth, a parking lane, or a rail platform for loading/unloading operations become free, respectively. The number of berths, parking lanes, and rail platforms is denoted by N_{b}, N_{p}, and N_{r}, respectively. The container queues are denoted by x_i^{b} ($i = 1, 2, \ldots, 2N_{\text{b}}$), x_i^{p} ($i = 1, 2, \ldots, 2N_{\text{p}}$), x_i^{r} ($i = 1, 2, \ldots, 2N_{\text{r}}$), and x_i^{y} ($i = 1, 2, \ldots, 18$), where the superscript "y" refers to "yard".

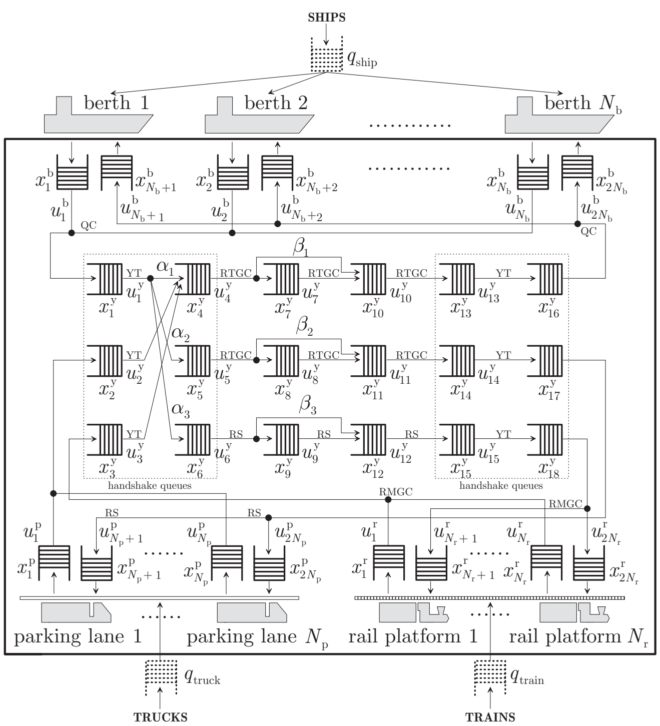

Fig. 1. Queueing model of a container terminal.

The containers to be unloaded from calling ships are in the queues x_i^{b} ($i = 1, 2, \ldots, N_{\text{b}}$). Each queue corresponds to a berth, and is served by quay cranes (QCs). Every time a QC lowers a container down to the quay, there must be a yard truck (YT) that receives it and promptly takes it to the rendezvous with the planned rubber tyred gantry crane (RTGC) in the import area of the storage yard, ready to pick it up and place it in the yard. This corresponds to the path from x_1^{y} to x_4^{y}, x_5^{y}, and x_6^{y}. Note that ideally there should be no waiting times in the handshake queues[1] x_1^{y}, x_4^{y}, x_5^{y}, and x_6^{y}. Similarly, using reach stackers (RSs) and rail mounted gantry cranes (RMGCs), the import flows from trucks and trains are modeled by the queues x_i^{p} ($i = 1, 2, \ldots, N_{\text{p}}$), x_2^{y}, x_4^{y}, and x_i^{r} ($i = 1, 2, \ldots, N_{\text{r}}$), x_3^{y}, x_4^{y}, respectively. The presence of containers within the yard is represented by means of the queues x_7^{y}, x_8^{y}, x_9^{y}, x_{10}^{y}, x_{11}^{y}, and x_{12}^{y}. Such queues refer to the six areas in which the yard is logically divided (i.e., import and export areas for ships, trucks, and trains). The flows from x_7^{y} to x_{10}^{y}, x_8^{y} to x_{11}^{y}, and x_9^{y} to x_{12}^{y} represent the rehandling activities (i.e., the movements of containers from the import area to the export area). The export flows are modeled by the couples of queues: x_{13}^{y} and x_{16}^{y} for ships; x_{14}^{y} and x_{17}^{y} for trucks; x_{15}^{y} and x_{18}^{y} for trains. The storage before the final delivery is made in the queues $x_{N_{\text{b}}+i}^{\text{b}}$ ($i = 1, 2, \ldots, N_{\text{b}}$) for ships, $x_{N_{\text{p}}+i}^{\text{p}}$ ($i = 1, 2, \ldots, N_{\text{p}}$) for trucks, and $x_{N_{\text{r}}+i}^{\text{r}}$ ($i = 1, 2, \ldots, N_{\text{r}}$) for trains.

The set of all the parameters and the variables of the model are briefly described in the following. We shall adopt a discrete-time setting with sampling time ΔT.

The state vector $\underline{x}(t) \triangleq \left[x_1^{\text{b}}(t), \ldots, x_{2N_{\text{b}}}^{\text{b}}(t), x_1^{\text{p}}(t), \ldots, x_{2N_{\text{p}}}^{\text{p}}(t), x_1^{\text{r}}(t), \ldots, x_{2N_{\text{r}}}^{\text{r}}(t), x_1^{\text{y}}(t), \ldots, x_{18}^{\text{y}}(t)\right]^{\text{T}}$ represents the queue lengths of containers waiting to be processed at time $t = 0, 1, \ldots$. Each component of the vector $\underline{x}(t)$ is greater than zero or equal to zero. We shall measure the lengths of the queues in TEU (a TEU corresponds to a typical 20-foot-long maritime container).

The control input vector $\underline{u}(t) \triangleq \left[u_1^{\text{b}}(t), \ldots, u_{2N_{\text{b}}}^{\text{b}}(t), u_1^{\text{p}}(t), \ldots, u_{2N_{\text{p}}}^{\text{p}}(t), u_1^{\text{r}}(t), \ldots, u_{2N_{\text{r}}}^{\text{r}}(t), u_1^{\text{y}}(t), \ldots, u_{15}^{\text{y}}(t)\right]^{\text{T}}$ is the collection of the percentages of the server capacities used for container transfers at time $t = 0, 1, \ldots$. Each component of the vector $\underline{u}(t)$ lies in the range $[0, 1]$.

The exogenous inputs $a_i^{\text{b}}(t) \geq 0$ ($i = 1, 2, \ldots, N_{\text{b}}$), $a_j^{\text{p}}(t) \geq 0$ ($j = 1, 2, \ldots, N_{\text{p}}$), and $a_l^{\text{r}}(t) \geq 0$ ($l = 1, 2, \ldots, N_{\text{r}}$) (in TEU) are the number of containers that enters the terminal at time $t = 0, 1, \ldots$ via ships, trucks, and trains, at the berth i, the parking lane j, and the rail platform l, respectively.

The parameters $\mu_i^{\text{b}}(t) \geq 0$ ($i = 1, 2, \ldots, 2N_{\text{b}}$), $\mu_i^{\text{p}}(t) \geq 0$ ($i = 1, 2, \ldots, 2N_{\text{p}}$), $\mu_i^{\text{r}}(t) \geq 0$ ($i = 1, 2, \ldots, 2N_{\text{r}}$), and $\mu_i^{\text{y}}(t) \geq 0$ ($i = 1, 2, \ldots, 15$) are the maximum container handling capacities for the various queues (in TEU/h) at time $t = 0, 1, \ldots$. Such parameters are related to the number of available resources and to the corresponding handling rates of each resource. Toward this end, we assume that each class of resource is composed of transfer machines with the same ideal time-varying capacity given by $r_{\text{QC}}(t)$ for QCs, $r_{\text{RMGC}}(t)$ for RMGCs, $r_{\text{RS}}(t)$ for RSs, $r_{\text{RTGC}}(t)$ for

[1] Following the approach of [5], a queue can be classified as either a real one or a "handshake" one. In the first case, the actual consignment in some area is provided, while in the second one there is an on-going transfer without storage. A real queue corresponds to the occupancy of spaces in the terminal. A handshake queue describes the delay that may occur when containers are transferred from one resource to another (i.e., the path from QCs to the storage yard via YTs). Ideally, the length of an handshake queue should be kept equal to zero since the higher the lengths of the handshake queues, the less efficient the terminal operations.

RTGCs, and $r_{\rm YT}(t)$ for YTs. The total numbers of such transfer machines are given by $n_{\rm QC}$, $n_{\rm RMGC}$, $n_{\rm RS}$, $n_{\rm RTGC}$, and $n_{\rm YT}$, respectively. Thus, the following relationships hold:

$$\mu_i^{\rm b}(t) = r_{\rm QC}(t)\, n_{\rm QC}, \quad i = 1, 2, \ldots, 2N_{\rm b} \tag{1a}$$
$$\mu_i^{\rm p}(t) = r_{\rm RS}(t)\, n_{\rm RS}, \quad i = 1, 2, \ldots, 2N_{\rm p} \tag{1b}$$
$$\mu_i^{\rm r}(t) = r_{\rm RMGC}(t)\, n_{\rm RMGC}, \quad i = 1, 2, \ldots, 2N_{\rm r} \tag{1c}$$
$$\mu_i^{\rm y}(t) = r_{\rm YT}(t)\, n_{\rm YT}, \quad i = 1, 2, 3, 13, 14, 15 \tag{1d}$$
$$\mu_i^{\rm y}(t) = r_{\rm RTGC}(t)\, n_{\rm RTGC}, \quad i = 4, 5, 7, 8, 10, 11 \tag{1e}$$
$$\mu_i^{\rm y}(t) = r_{\rm RS}(t)\, n_{\rm RS}, \quad i = 6, 9, 12 \tag{1f}$$

where $t = 0, 1, \ldots$.

The vector $\underline{y}(t) \triangleq \big[y_1^{\rm b}(t), \ldots, y_{N_{\rm b}}^{\rm b}(t), y_1^{\rm p}(t), \ldots, y_{N_{\rm p}}^{\rm p}(t), y_1^{\rm r}(t), \ldots, y_{N_{\rm r}}^{\rm r}(t)\big]^{\rm T} \in \{0, 1\}^{N_{\rm b}+N_{\rm p}+N_{\rm r}}$ is the collection of the binary variables that are used to model the departure or stay of a carrier depending on the fact that the planned loading/unloading operations are finished or not at time $t = 0, 1, \ldots$. For instance, $y_i^{\rm b}(t)$ is equal to 0 when the ship at the berth i has finished all the loading/unloading operations and therefore can leave the terminal. Otherwise it is equal to 1. Similarly for the binary variables $y_i^{\rm p}(t)$ ($i = 1, 2, \ldots, N_{\rm p}$) and $y_i^{\rm r}(t)$ ($i = 1, 2, \ldots, N_{\rm r}$). In other words, the variables $y_i^{\rm b}(t)$ ($i = 1, 2, \ldots, N_{\rm b}$), $y_i^{\rm p}(t)$ ($i = 1, 2, \ldots, N_{\rm p}$), and $y_i^{\rm r}(t)$ ($i = 1, \ldots, N_{\rm r}$) assume their values as follows:

$$y_i^{\rm w}(t) = \begin{cases} 0 \text{ if } x_{N_{\rm w}+i}^{\rm w}(t) + \Delta T \mu_{N_{\rm w}+i}^{\rm w}(t) u_{N_{\rm w}+i}^{\rm w}(t) = s_i^{\rm w}(t) \\ \quad \text{and } x_i^{\rm w}(t) = 0 \\ 1 \text{ otherwise} \end{cases}$$
$$i = 1, 2, \ldots, N_{\rm w}, \ \ {\rm w} = {\rm b}, {\rm p}, {\rm r} \tag{2}$$

where $t = 0, 1, \ldots$ and $s_i^{\rm w}(t)$ ($i = 1, 2, \ldots, N_{\rm w}$, ${\rm w} = {\rm b}, {\rm p}, {\rm r}$) is defined as the quantity of containers scheduled, at time t, for loading before departure with the carrier at the berth, the parking lane, or the rail platform i. The introduction of such binary variables allows one to have at disposal a more precise, though more complex, model with respect to that reported in [5].

The parameters $\alpha_i(t) \geq 0$ ($i = 1, 2, 3$) are the sharing percentages of import traffic at time $t = 0, 1, \ldots$: $\alpha_1(t)$ for transhipment, $\alpha_2(t)$ for ship to truck, and $\alpha_3(t)$ for ship to rail. Clearly, the equality $\alpha_1(t) + \alpha_2(t) + \alpha_3(t) = 1$ holds for all $t = 0, 1, \ldots$. The parameters $\beta_i(t) \geq 0$ ($i = 1, 2, 3$) are the percentages of containers that are ready for prompt export operations, and thus can skip the rehandling procedure at time $t = 0, 1, \ldots$.

The dynamics that results from the balance of input and output container flows for all the queues in Fig. 1 is given by the following equations:

$$x_i^{\rm w}(t+1) = x_i^{\rm w}(t) + a_i^{\rm w}(t) - \Delta T \mu_i^{\rm w}(t) u_i^{\rm w}(t), \quad i = 1, 2, \ldots, N_{\rm w}, \ \ {\rm w} = {\rm b}, {\rm p}, {\rm r} \tag{3a}$$
$$x_i^{\rm y}(t+1) = x_i^{\rm y}(t) + \Delta T \left[\sum_{j=1}^{N_{\rm w}} \mu_j^{\rm w}(t) u_j^{\rm w}(t) - \mu_i^{\rm y}(t) u_i^{\rm y}(t) \right], \quad i = 1, 2, 3, \ \ {\rm w} = {\rm b}, {\rm p}, {\rm r} \tag{3b}$$
$$x_4^{\rm y}(t+1) = x_4^{\rm y}(t) + \Delta T [\alpha_1(t) \mu_1^{\rm y}(t) u_1^{\rm y}(t) + \mu_2^{\rm y}(t) u_2^{\rm y}(t) + \mu_3^{\rm y}(t) u_3^{\rm y}(t) - \mu_4^{\rm y}(t) u_4^{\rm y}(t)] \tag{3c}$$
$$x_i^{\rm y}(t+1) = x_i^{\rm y}(t) + \Delta T \left[\alpha_{i-3}(t) \mu_1^{\rm y}(t) u_1^{\rm y}(t) - \mu_i^{\rm y}(t) u_i^{\rm y}(t)\right], \quad i = 5, 6 \tag{3d}$$
$$x_i^{\rm y}(t+1) = x_i^{\rm y}(t) + \Delta T \left\{\mu_{i-3}^{\rm y}(t) u_{i-3}^{\rm y}(t) [1 - \beta_{i-6}(t)] - \mu_i^{\rm y}(t) u_i^{\rm y}(t)\right\}, \quad i = 7, 8, 9 \tag{3e}$$
$$x_i^{\rm y}(t+1) = x_i^{\rm y}(t) + \Delta T \left[\mu_{i-6}^{\rm y}(t) u_{i-6}^{\rm y}(t) \beta_{i-9}(t) + \mu_{i-3}^{\rm y}(t) u_{i-3}^{\rm y}(t) - \mu_i^{\rm y}(t) u_i^{\rm y}(t)\right], \quad i = 10, 11, 12 \tag{3f}$$
$$x_i^{\rm y}(t+1) = x_i^{\rm y}(t) + \Delta T \big[\mu_{i-3}^{\rm y}(t) u_{i-3}^{\rm y}(t) - \mu_i^{\rm y}(t) u_i^{\rm y}(t)\big], \quad i = 13, 14, 15 \tag{3g}$$
$$x_i^{\rm y}(t+1) = x_i^{\rm y}(t) + \Delta T \left[\mu_{i-3}^{\rm y}(t) u_{i-3}^{\rm y}(t) - \sum_{j=1}^{N_{\rm w}} \mu_{N_{\rm w}+j}^{\rm w}(t) u_{N_{\rm w}+j}^{\rm w}(t)\right], \quad i = 16, 17, 18, \ \ {\rm w} = {\rm b}, {\rm p}, {\rm r} \tag{3h}$$
$$x_{N_{\rm w}+i}^{\rm w}(t+1) = y_i^{\rm w}(t) \big[x_{N_{\rm w}+i}^{\rm w}(t) + \Delta T \mu_{N_{\rm w}+i}^{\rm w}(t) u_{N_{\rm w}+i}^{\rm w}(t)\big], \quad i = 1, 2, \ldots, N_{\rm w}, \ \ {\rm w} = {\rm b}, {\rm p}, {\rm r} \tag{3i}$$

where $t = 0, 1, \ldots$. To account for the boundedness and sharing of resources, we need to impose constraints on the control variables $\underline{u}(t)$ as follows:

$$\sum_{j=1}^{2N_{\rm b}} u_j^{\rm b}(t) \leq 1 \tag{4a}$$
$$\sum_{j=1}^{2N_{\rm p}} u_j^{\rm p}(t) + u_6^{\rm y}(t) + u_9^{\rm y}(t) + u_{12}^{\rm y}(t) \leq 1 \tag{4b}$$
$$\sum_{j=1}^{2N_{\rm r}} u_j^{\rm r}(t) \leq 1 \tag{4c}$$
$$u_4^{\rm y}(t) + u_5^{\rm y}(t) + u_7^{\rm y}(t) + u_8^{\rm y}(t) + u_{10}^{\rm y}(t) + u_{11}^{\rm y}(t) \leq 1 \tag{4d}$$
$$u_1^{\rm y}(t) + u_2^{\rm y}(t) + u_3^{\rm y}(t) + u_{13}^{\rm y}(t) + u_{14}^{\rm y}(t) + u_{15}^{\rm y}(t) \leq 1 \tag{4e}$$

where $t = 0, 1, \ldots$. In order to account also for the limits of the yard, we have to impose the following bound on the state variables:

$$\sum_{j=7}^{12} x_j^{\rm y}(t) \leq b \tag{5}$$

where $t = 0, 1, \ldots$ and b is a constant that represents the space limitation of the yard (in TEU).

As regards the rehandling of containers from the import area to the export area, a common solution consists in using the resources that are not employed in other transfer operations. A model to account for such a rehandling strategy is the following:

$$u_7^{\rm y}(t) = \min\left\{ \frac{x_7^{\rm y}(t) + \Delta T \mu_4^{\rm y}(t) u_4^{\rm y}(t) [1 - \beta_1(t)]}{\Delta T \mu_7^{\rm y}(t)}, \gamma(t) [1 - u_4^{\rm y}(t) - u_5^{\rm y}(t) - u_{10}^{\rm y}(t) - u_{11}^{\rm y}(t)] \right\} \tag{6a}$$

$$u_8^{\mathrm{y}}(t) = \min\left\{ \frac{x_8^{\mathrm{y}}(t) + \Delta T \mu_5^{\mathrm{y}}(t) u_5^{\mathrm{y}}(t)[1-\beta_2(t)]}{\Delta T \mu_8^{\mathrm{y}}(t)}, \right. \left. [1-\gamma(t)][1-u_4^{\mathrm{y}}(t)-u_5^{\mathrm{y}}(t)-u_{10}^{\mathrm{y}}(t)-u_{11}^{\mathrm{y}}(t)] \right\} \tag{6b}$$

$$u_9^{\mathrm{y}}(t) = \min\left\{ \frac{x_9^{\mathrm{y}}(t) + \Delta T \mu_6^{\mathrm{y}}(t) u_6^{\mathrm{y}}(t)[1-\beta_3(t)]}{\Delta T \mu_9^{\mathrm{y}}(t)}, \right. \left. 1-\sum_{i=1}^{2N_{\mathrm{p}}} u_i^{\mathrm{p}}(t) - u_6^{\mathrm{y}}(t) - u_{12}^{\mathrm{y}}(t) \right\} \tag{6c}$$

where $t = 0, 1, \ldots$ and $\gamma(t) \in (0, 1)$ represents the percentage of unused RTGC transfer capacity that is employed for the rehandling of containers from ships. The first arguments of the minimization operators ensure to keep the positivity of the involved state variables, while the second ones are the percentages of unused resources. It is noteworthy that the presence of the minimization operators introduces nonsmooth nonlinearities into the model.

III. Predictive control of container flows

In this section, we shall use the model (1)-(6) to develop an optimal resource allocation strategy based on predictive control. The goal of predictive control is to allocate the resources in order to optimize an objective function that is related to the performance of the terminal over a certain time horizon from the current instant. Toward this end, we assume that the quantities of containers scheduled for loading/unloading on the arrival of carriers are known.

In developing and analyzing the performances of any control strategy, we shall account for the following requirements: (i) the departure of a carrier occurs only after the completion of the scheduled loading and unloading plan; (ii) the handshake queues are empty; (iii) the lay time of carriers (i.e., the time needed to complete the loading/unloading operations) is as small as possible.

As regards the requirement (i), in order to ensure that a carrier leaves the terminal only when all the export operations are completed, we need to impose the following constraints:

$$x_{N_{\mathrm{w}}+i}^{\mathrm{w}}(t) + \Delta T \mu_{N_{\mathrm{w}}+i}^{\mathrm{w}}(t) u_{N_{\mathrm{w}}+i}^{\mathrm{w}}(t) < s_i^{\mathrm{w}}(t) + M[1-y_i^{\mathrm{w}}(t)] \quad i = 1, 2, \ldots, N_{\mathrm{w}}, \ \mathrm{w} = \mathrm{b}, \mathrm{p}, \mathrm{r} \tag{7a}$$

$$x_{N_{\mathrm{w}}+i}^{\mathrm{w}}(t) + \Delta T \mu_{N_{\mathrm{w}}+i}^{\mathrm{w}}(t) u_{N_{\mathrm{w}}+i}^{\mathrm{w}}(t) \le s_i^{\mathrm{w}}(t) + M y_i^{\mathrm{w}}(t) \quad i = 1, 2, \ldots, N_{\mathrm{w}}, \ \mathrm{w} = \mathrm{b}, \mathrm{p}, \mathrm{r} \tag{7b}$$

$$x_{N_{\mathrm{w}}+i}^{\mathrm{w}}(t) + \Delta T \mu_{N_{\mathrm{w}}+i}^{\mathrm{w}}(t) u_{N_{\mathrm{w}}+i}^{\mathrm{w}}(t) \ge s_i^{\mathrm{w}}(t) - M y_i^{\mathrm{w}}(t) \quad i = 1, 2, \ldots, N_{\mathrm{w}}, \ \mathrm{w} = \mathrm{b}, \mathrm{p}, \mathrm{r} \tag{7c}$$

where $t = 0, 1, \ldots$ and M is a very large positive constant. Constraint (7a) is active if $y_i^{\mathrm{w}}(t)$ is equal to 1, i.e., if the loading operations are not completed, and hence the carrier is not ready for departure. The effect of such a constraint is that of avoiding loading into the carrier more containers than those that are requested. The inequality constraints (7b) and (7c) provide an equality constraint if they are simultaneously activated with $y_i^{\mathrm{w}}(t)$ equal to 0 (i.e., when the loaded containers are equal to those that are requested).

In order to obtain the conclusion of all the import operations before departure, we need to introduce another constraint:

$$x_i^{\mathrm{w}}(t) \le M\, y_i^{\mathrm{w}}(t)\ , \ i = 1, \ldots, N_{\mathrm{w}}, \ \mathrm{w} = \mathrm{b}, \mathrm{p}, \mathrm{r} \tag{8}$$

where $t = 0, 1, \ldots$ and M is the previously introduced large positive constant. For example, when $y_i^{\mathrm{w}}(t) = 0$, the constraint (8) can be satisfied only if $x_i^{\mathrm{w}}(t) = 0$, i.e., if all the containers to be unloaded have been unloaded and hence the carrier can leave the terminal. By contrast, if $y_i^{\mathrm{w}}(t) = 1$, (8) is trivially satisfied.

We can exactly impose the condition in (ii) via the following constraints:

$$\sum_{j=1}^{N_{\mathrm{w}}} \mu_j^{\mathrm{w}}(t) u_j^{\mathrm{w}}(t) - \mu_i^{\mathrm{y}}(t) u_i^{\mathrm{y}}(t) = 0, \quad i = 1, 2, 3, \ \mathrm{w} = \mathrm{b}, \mathrm{p}, \mathrm{r} \tag{9a}$$

$$\alpha_1(t)\mu_1^{\mathrm{y}}(t)u_1^{\mathrm{y}}(t) + \mu_2^{\mathrm{y}}(t)u_2^{\mathrm{y}}(t) + \mu_3^{\mathrm{y}}(t)u_3^{\mathrm{y}}(t) - \mu_4^{\mathrm{y}}(t)u_4^{\mathrm{y}}(t) = 0 \tag{9b}$$

$$\alpha_{i-3}(t)\mu_1^{\mathrm{y}}(t)u_1^{\mathrm{y}}(t) - \mu_i^{\mathrm{y}}(t)u_i^{\mathrm{y}}(t) = 0, \quad i = 5, 6 \tag{9c}$$

$$\mu_i^{\mathrm{y}}(t)u_i^{\mathrm{y}}(t) - \mu_{i+3}^{\mathrm{y}}(t)u_{i+3}^{\mathrm{y}}(t) = 0, \quad i = 10, 11, 12 \tag{9d}$$

$$\mu_i^{\mathrm{y}}(t)u_i^{\mathrm{y}}(t) - \sum_{j=1}^{N_{\mathrm{w}}} \mu_{N_{\mathrm{w}}+j}^{\mathrm{w}}(t) u_{N_{\mathrm{w}}+j}^{\mathrm{w}}(t) = 0, \quad i = 13, 14, 15, \ \mathrm{w} = \mathrm{b}, \mathrm{p}, \mathrm{r} \tag{9e}$$

where $t = 0, 1, \ldots$.

We can take into consideration the requirement (iii) via a suitable choice of a cost function to be minimized. Toward this end, we refer to a function h (in general nonlinear), which provides a measure of performance given by $h\left[\underline{x}(t), \underline{y}(t), \underline{u}(t)\right]$.

We chose the minimization of the lay times of carriers as a management goal since a carrier that remains for too long in the terminal results in higher costs for both a useless employment of resources and a poor service to customers. As in [5], we focused on the following function h:

$$h\left[\underline{x}(t), \underline{y}(t), \underline{u}(t)\right] = \sum_{\mathrm{w}\in\{\mathrm{b},\mathrm{p},\mathrm{r}\}} \sum_{j=1}^{N_{\mathrm{w}}} \left\{ c_1^{\mathrm{w}} p_j^{\mathrm{w}}(t)[1-u_j^{\mathrm{w}}(t)]\, y_j^{\mathrm{w}}(t) + c_2^{\mathrm{w}} p_j^{\mathrm{w}}(t)[1-u_{N_{\mathrm{w}}+j}^{\mathrm{w}}(t)]\, y_j^{\mathrm{w}}(t) \right\} \tag{10}$$

where the parameters c_1^{w} and c_2^{w} $(\mathrm{w} = \mathrm{b}, \mathrm{p}, \mathrm{r})$ are positive constants and $p_1^{\mathrm{w}}(t), \ldots, p_{N_{\mathrm{w}}}^{\mathrm{w}}(t)$ $(\mathrm{w} = \mathrm{b}, \mathrm{p}, \mathrm{r})$ are "priority" terms that allows one to serve the carriers "fairly" over time. At each time $t = 0, 1 \ldots$, a higher priority is assigned to the carriers that have been in the terminal for a longer time by increasing the weights of the corresponding terms in the cost. A performance index based on (10) aims to minimize the lay times of ships, trucks, and trains. Indeed, since the time a carrier spends in the terminal depends tightly on how fast the containers to be loaded to and unloaded from the carriers are moved, one can reduce the overall delay by keeping the control inputs as higher as possible.

A predictive control approach consists in minimizing, at each time $t = 0, 1, \ldots$, and under the various constraints, a cost function of the form

$$J_t[\underline{x}(t), \underline{y}(t, t+T-1), \underline{u}(t, t+T-1)] = \sum_{k=t}^{t+T-1} h[\underline{x}(k), \underline{y}(k), \underline{u}(k)] \quad (11)$$

where T is the length of the forward horizon and $\underline{y}(t, t+T-1)$ and $\underline{u}(t, t+T-1)$ stand for $\underline{y}(t), \ldots, \underline{y}(t+T-1)$ and $\underline{u}(t), \ldots, \underline{u}(t+T-1)$, respectively.

Summing up, given $\underline{x}(t)$ at each time $t = 0, 1, \ldots$, we need to solve the following optimization problem:

$$\min_{\substack{\underline{u}(t,t+T-1)\\ \underline{y}(t,t+T-1)}} J_t[\underline{x}(t), \underline{y}(t, t+T-1), \underline{u}(t, t+T-1)] \quad (12)$$

subject to the constraints (1), r.h.s. of (3)≥ 0, (4), (5), (7)-(9), $\underline{u}(k) \in [0,1]^{2N_{\mathrm{b}}+2N_{\mathrm{p}}+2N_{\mathrm{r}}+15}$, $\underline{y}(k) \in \{0,1\}^{N_{\mathrm{b}}+N_{\mathrm{p}}+N_{\mathrm{r}}}$, and $u_7^{\mathrm{y}}(k) = u_8^{\mathrm{y}}(k) = u_9^{\mathrm{y}}(k) = 0$ for $k = t, \ldots, t+T-1$. Once the optimal controls for $\underline{y}(k)$ and $u_i^{\mathrm{b}}(k)$ $(i = 1, 2, \ldots, 2N_{\mathrm{b}})$, $u_i^{\mathrm{p}}(k)$ $(i = 1, 2, \ldots, 2N_{\mathrm{p}})$, $u_i^{\mathrm{r}}(k)$ $(i = 1, 2, \ldots, 2N_{\mathrm{r}})$, and $u_i^{\mathrm{y}}(k)$ $(i = 1, 2, \ldots, 6, 10, 11, \ldots, 15)$ have been found for $k = t, \ldots, t+T-1$, only the first control action is retained and applied, and unused resources are employed for the rehandling operations, i.e., by selecting $u_7^{\mathrm{y}}(t)$, $u_8^{\mathrm{y}}(t)$, and $u_9^{\mathrm{y}}(t)$ according to (6).

A possible approach to find a solution to problem (12) is based on Branch-and-Bound Mixed-Integer NonLinear Programming techniques (we shall refer to such methodologies as BBMINLP) because of the binary variables that are involved in both the state equation and the constraints. This complicates the process of finding a solution, thus a technique that avoids using them is highly desirable. In fact, we can avoid dealing with binary variables by treating the constraints (3i) as nonsmooth step functions. More specifically, (3i) can be rewritten as follows:

$$x_{N_{\mathrm{w}}+i}^{\mathrm{w}}(t+1) = \begin{cases} 0 & \text{if} \quad x_i^{\mathrm{w}}(t) = 0 \quad \text{and} \quad x_{N_{\mathrm{w}}+i}^{\mathrm{w}}(t) \\ & \quad + \Delta T \mu_{N_{\mathrm{w}}+i}^{\mathrm{w}}(t) u_{N_{\mathrm{w}}+i}^{\mathrm{w}}(t) = s_i^{\mathrm{w}}(t) \\ x_{N_{\mathrm{w}}+i}^{\mathrm{w}}(t) + \Delta T \mu_{N_{\mathrm{w}}+i}^{\mathrm{w}}(t) u_{N_{\mathrm{w}}+i}^{\mathrm{w}}(t) & \text{otherwise} \end{cases}$$
$$i = 1, 2, \ldots, N_{\mathrm{w}}, \;\; \mathrm{w} = \mathrm{b}, \mathrm{p}, \mathrm{r}.$$

Such constraints can be expressed by the following relationships:

$$x_{N_{\mathrm{w}}+i}^{\mathrm{w}}(t+1) = \left[x_{N_{\mathrm{w}}+i}^{\mathrm{w}}(t) + \Delta T \mu_{N_{\mathrm{w}}+i}^{\mathrm{w}}(t) u_{N_{\mathrm{w}}+i}^{\mathrm{w}}(t)\right] \chi\Big[s_i^{\mathrm{w}}(t) - x_{N_{\mathrm{w}}+i}^{\mathrm{w}}(t) - \Delta T \mu_{N_{\mathrm{w}}+i}^{\mathrm{w}}(t) u_{N_{\mathrm{w}}+i}^{\mathrm{w}}(t) + x_i^{\mathrm{w}}(t)\Big],$$
$$i = 1, 2, \ldots, N_{\mathrm{w}}, \;\; \mathrm{w} = \mathrm{b}, \mathrm{p}, \mathrm{r}$$

where $\chi(\cdot)$ is the step function (i.e., $\chi(z) = 1$ if $z > 0$, $\chi(z) = 0$ otherwise). In this way, we avoid using the binary variables, and the problem reduces to a nonlinear programming one with no integer variables but with nonsmooth functions, thus resorting for the solution to Real NonLinear Programming techniques (we shall refer to such methodologies as RNLP). Unfortunately, it may be quite demanding to find a solution of RNLP and particularly BBMINLP problems in a real-time context, especially with predictive horizons larger than $T = 1$ and a large number of decision and state variables. Following the RNLP approach, the binary variables of $\underline{y}(t)$ are replaced with the corresponding functions $\chi(\cdot)$ and the predictive control problem (12) reduces to find only the optimal inputs $\underline{u}^{\circ}(t, t+T-1)$. Since only the first control input (i.e., $\underline{u}^{\circ}(t)$) is applied, a different approach is proposed that consists in approximating off line the optimal feedback control function $\underline{x}(t) \mapsto \underline{u}^{\circ}(t)$ by means of some approximators in order to generate the control action on line almost instantaneously. In particular, once T has been selected, we can (i) solve off line many RNLP problems starting with different initial conditions, (ii) collect the pairs $(\underline{x}(t), \underline{u}^{\circ}(t))$, and (iii) apply some regression method to approximate such pairs. Toward this end, we shall employ one-hidden-layer feedforward neural networks (OHLFFNNs) with sigmoidal activation function in the hidden layer. Thus, the regression problem becomes a neural learning task to find the optimal weights that allow to compute the control action on line via the trained OHLFFNN.

It is worth noting that the resulting approximate control law may not satisfy exactly some constraints. However, to deal with this difficulty, some simple solutions can be devised. For example, if the equality constraints in (9) were not satisfied, we could simply avoid moving from queues containers that would violate the flow balance. Concerning the inequality constraints (4), we can normalize the outputs of the neural networks in such a way to impose their satisfaction. Clearly, the more precise the approximation, the smaller the correction. Thus, in order to ensure a desired precision, OHLFFNNs of sufficiently large size should be used to take on the structure of the approximating function that generates the control.

IV. Numerical results

In this section, we present the simulation results obtained in a case study that refers to a medium-size container terminal in North-West Italy. Details on the characteristics of the terminal as to layout, interarrival times of carriers, inbound and outbound container flows, and capacities of the various transfer machines can be found in [5].

The solutions of the various problems (12) at $t = 0, 1, \ldots$ were found using either Matlab standard nonlinear programming techniques without using the derivatives of both the cost function and the constraints according to a RNLP approach, or a standard branch-and-bound algorithm to perform a BBMINLP optimization. As regards the neural approach, we used a OHLFFNN with 10 sigmoidal activation functions in the hidden layer. The neural training was done using a data set made up of 1500 state-control pairs obtained by the RNLP approach. The selection of the optimal weights was performed with the Levenberg-Marquardt training algorithm available in Matlab.

The performances of the terminal were evaluated by solving the optimal control problem (12) for different predictive horizons T and computing the number of served ships, trucks, and trains, the quantity of transferred containers,

TABLE I

PERFORMANCES GIVEN BY THE BBMINLP, RNLP, AND OHLFFNN APPROACHES FOR DIFFERENT CONTROL HORIZONS.

	Control horizon	$T = 1$			$T = 2$			$T = 3$		
	Carrier type	Ship	Truck	Train	Ship	Truck	Train	Ship	Truck	Train
BBMINLP	Number of served carriers	6	71	13	6	72	13	6	72	13
	Transferred containers [TEU]	4096	4860	2110	4096	4866	2110	4096	4866	2110
	Mean lay time [h]	13.2	0.33	3.76	13.1	0.33	3.80	12.5	0.35	3.90
	I_{T_f}	$183.8 \cdot 10^9$			$180.2 \cdot 10^9$			$167.4 \cdot 10^9$		
	Mean on-line computation time [min]	7.6			49.0			154.1		
RNLP	Number of served carriers	6	71	13	6	70	13	6	72	13
	Transferred containers [TEU]	4096	4860	2110	4096	4850	2110	4096	4866	2110
	Mean lay time [h]	13.6	0.32	3.84	13.5	0.33	3.84	12.8	0.36	3.88
	I_{T_f}	$209.1 \cdot 10^9$			$205.5 \cdot 10^9$			$178.6 \cdot 10^9$		
	Mean on-line computation time [min]	0.16			0.79			3.5		
OHLFFNN	Number of served carriers	6	66	13	6	68	13	6	67	13
	Transferred containers [TEU]	4096	4746	2110	4096	4810	2110	4096	4780	2110
	Mean lay time [h]	15.3	0.50	4.75	14.8	0.45	4.42	13.7	0.45	4.65
	I_{T_f}	$240.0 \cdot 10^9$			$206.8 \cdot 10^9$			$181.6 \cdot 10^9$		
	Mean on-line computation time [min]	0.0011			0.0011			0.0011		

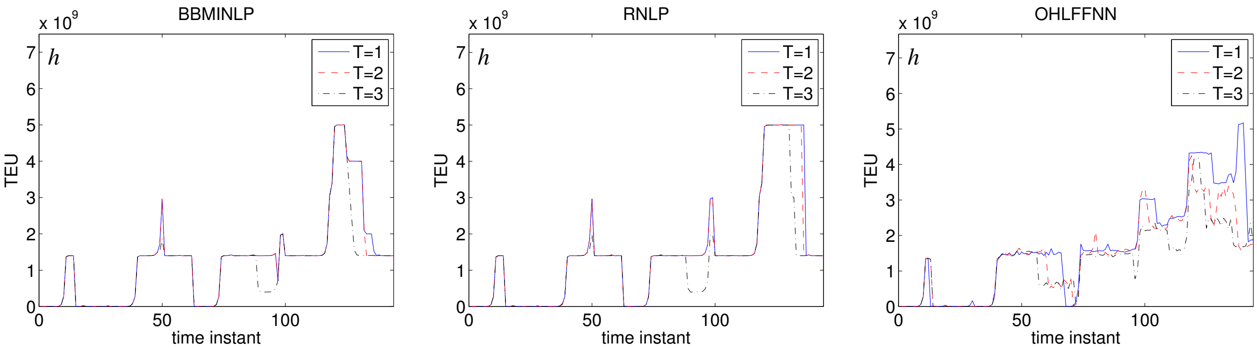

Fig. 2. Values of the function h given by the BBMINLP, RNLP, and OHLFFNN approaches for different control horizons.

the mean lay times of carriers, and the index $I_{T_f} = \sum_{t=0}^{T_f} h\left[\underline{x}(k), \underline{y}(k), \underline{u}(k)\right]$ where T_f is the simulation length. In the results reported in Table I, T_f was chosen equal to 144, which corresponds to a simulation length of 72 h. The simulations were performed on a 3.2 GHz Pentium 4 PC with 1 GB of RAM.

As expected, the longer the control horizon T, the lower the value of the function h, as pictorially shown in Fig. 2 for the BBMINLP, RNLP, and OHLFFNN approaches. As shown in Table I, BBMINLP provides better results in terms of the final minimization cost, while RNLP demands a reduced computational burden with respect to BBMINLP; this reduction is paid with a small decay of performances. The OHLFFNN control performs worse than the other approaches, but it has the great advantage of a very low on-line computational burden. Moreover, in Fig. 2 note that BBMINLP ensures better performances in the transient with respect to RNLP, while the behavior of the two approaches is practically the same when the values of the function h are constant. The OHLFFNN control performs similarly to the controls obtained by BBMINLP and RNLP for $t \in [0, 100]$, with a small decay of performances for $t \in [100, 144]$.

REFERENCES

[1] K. M. Van Hee and R. J. Wijbrands, “Decision support system for container terminal planning,” *European J. of Operational Research*, vol. 34, no. 3, pp. 262–272, 1988.

[2] E. Kozan, “Optmising container transfer at multimodal terminals,” *Math. and Computer Modelling*, vol. 31, no. 10-12, pp. 235–243, 2000.

[3] L. M. Gambardella, A. Mastrolilli, A. E. Rizzoli, and M. Zaffalon, “An optimization methodology for intermodal terminal management,” *J. of Intelligent Manufacturing*, vol. 12, no. 5-6, pp. 521–534, 2001.

[4] A. Alessandri, S. Sacone, and S. Siri, “Modelling and optimal receding-horizon control of maritime container terminals,” *J. of Mathematical Modelling and Algorithms*, vol. 6, no. 1, pp. 109–133, 2007.

[5] A. Alessandri, C. Cervellera, M. Cuneo, M. Gaggero, and G. Soncin, “Modeling and feedback control for resource allocation and performance analysis in container terminals,” *IEEE Trans. on Intelligent Transportation Systems*, to appear.

[6] D. Mayne, J. Rawlings, C. Rao, and P. Scokaert, “Constrained model predictive control: stability and optimality,” *Automatica*, vol. 36, no. 6, pp. 789–814, 2000.

[7] D. Li and X. Sun, *Nonlinear Integer Programming*. New York, NY: Springer, 2006.

[8] V. Kůrková, “Approximation of functions by perceptron networks with bounded number of hidden units,” *Neural Networks*, vol. 8, no. 5, pp. 745–750, 1995.

[9] A. Barron, “Universal approximation bounds for superpositions of a sigmoidal function,” *IEEE Trans. Inf. Theory*, vol. 39, no. 3, pp. 930–945, 1993.

[10] R. Zoppoli, M. Sanguineti, and T. Parisini, “Approximating networks and extended Ritz method for the solution of functional optimization problems,” *J. of Optim. Theory and Applications*, vol. 112, no. 2, pp. 403–439, 2002.

[11] V. Kůrková and M. Sanguineti, “Comparison of worst-case errors in linear and neural network approximation,” *IEEE Trans. Inf. Theory*, vol. 48, no. 1, pp. 264–275, 2002.