A distributed control architecture for logistics operations in flexible manufacturing systems

Francesco Giannini *Student Member , IEEE,* Domenico Famularo,
Giancarlo Fortino *Fellow, IEEE* and Giuseppe Franzè *Senior Member, IEEE.*

***Abstract*— In this paper, the problem of controlling autonomous vehicles in a Flexible Manufacturing System is addressed in order to optimize logistic operations. To this end, vehicles are required to navigate between machines and from/to the Load/Unload station. The core contribution of this paper is to propose a set-theoretic distributed Model Predictive Control in charge of controlling the autonomous vehicles properly integrated with a Reinforcement Learning scheme to address the routing problem. In addition, vehicles are organized as platoons in order to improve the efficiency of the overall architecture. The numerical simulation shows the effectiveness of the proposed approach.**

I. Introduction

In today's competitive economy, Flexible Manufacturing Systems (FMSs) [14] encounter new challenges in improving production efficiency and sustainability. A successful industrial answer stands in the massive use of integrated automation within industrial systems [4]. Inside this framework, advancements in Autonomous Vehicles (AVs) is fundamental because of their proved ability to optimize costs and improve societal outcomes as safety and accessibility [10] [7]. AVs present an innovative solution capable of efficiently managing production lines, warehouse inventories, and intra- as well as inter-logistics services across diverse economic sectors [13] [5].

The main contribution of this paper is the proposal of an integrated architecture combining a Reinforcement Learning scheme for routing decisions with a set-theoretic Receding Horizon Controller in charge of computing feasible control actions to be applied to the AVs. Moreover, in order to improve production efficiency, the vehicles are organized as a set of platoons.

The proposed approach needs to be initialized during the offline phase when (1) an external unit translates the FMS task into set-points to be assigned to the AVs' platoons and (2) safe state trajectory tube are computed in a set-theoretic Model Predictive Control fashion [6]. Then, our proposed unit computes routing actions and corresponding control moves to complete the prescribed task. Finally, it is worth emphasizing that an anti-collision procedure is proposed to address scenarios where two platoons travel in opposite directions along the same corridor.

Giuseppe Franzè is with DIMEG, Università della Calabria, Via Pietro Bucci, Cubo 42-C, Rende (CS), 87036, ITALY, giuseppe.franze@unical.it

Francesco Giannini, Domenico Famularo and Giancarlo Fortino and are with DIMES, Università della Calabria, Via Pietro Bucci, Cubo 41/42-C, Rende (CS), 87036, ITALY, francesco.giannini@dimes.unical.it {domenico.famularo,giancarlo.fortino}@unical.it

Notation and definitions

Given a set $\mathcal{A}$, $|\mathcal{A}|$ denotes its cardinality.

Definition 1: Let $G = (V, \mathcal{E})$ be a graph with $V := \{v_1, \dots, v_L\}$ nodes and $\mathcal{E}$ edges, i.e. $\mathcal{E} := \{(i,j) : i,j \in L,\ i \neq j\}$. A node $i \in V$ is connected (or adjacent) to a node $j \in L\backslash\{i\}$ if there exists an edge from i to j. □

Definition 2: A $m-$ length path of a given graph G is a sequence of m nodes $\{v_{i_1}, \dots, v_{i_m}\}$ such that for $k = 1, \dots, m$, the vertices v_{i_k} and $v_{i_{k+1}}$ are adjacent. □

Definition 3: A graph G is connected if there is a path for every pair of vertices $v_i,\ v_j \in V$, with $i \neq j$. □

Definition 4: Given a set $\mathcal{S} \subset \mathbb{R}^n$ and a point $p \in \mathbb{R}^n$, the norm-2 distance between p and $\mathcal{S}$ is defined as:

$$dist(p, \mathcal{S}) := \inf_{s \in \mathcal{S}} \|s - p\|_2,$$

Definition 5: Given two sets $\mathcal{S}_1, \mathcal{S}_2 \subset \mathbb{R}^n$, the norm-2 distance between $\mathcal{S}_1$ and $\mathcal{S}_2$ is defined as:

$$dist(\mathcal{S}_1, \mathcal{S}_2) := \inf \{\|s_1 - s_2\|_2 \ :\ s_1 \in \mathcal{S}_1,\ s_2 \in \mathcal{S}_2\}$$

Consider a discrete-time linear time-invariant system

$$x(t+1) = \Phi x(t) + Gu(t) \tag{1}$$

where $t \in \mathbb{Z}_+ := \{0, 1, ...\}$, $x(t) \in \mathbb{R}^n$ is the state, $u(t) \in \mathbb{R}^m$ the control input and

$$u(t) \in \mathcal{U}, \quad x(t) \in \mathcal{X}, \quad \forall t \geq 0, \tag{2}$$

with $\mathcal{U}$, $\mathcal{X}$ convex and compact subsets of $\mathbb{R}^m$ and $\mathbb{R}^n$, respectively.

Definition 6: A set $\Xi \subseteq \mathcal{X}$ is said to be Positively Invariant (PI) set for (1) under constraints (2) if there exists a control law $u := f(x(t)) \in \mathcal{U}$ such that $\forall\, x(0) \in \Xi \rightarrow \Phi x(t) + Gf(x(t)) \in \Xi,\ \forall t \in \mathbb{Z}_+$. □

Given the plant (1) it is possible to compute the sets of states i-step controllable to $\mathcal{T}_0 := \mathcal{T}$ via the following recursions [1]:

$$\mathcal{T}_i := \{x : \exists u \in \mathcal{U} : \Phi x + Gu \in \mathcal{T}_{i-1}\} \tag{3}$$

With $x(t+h|t)$ and $u(t+h|t)$ denote the $h-th$ predicted state and predicted input at time t, respectively.

Given a matrix $H \in \mathbb{R}^{n \times m}$, $col_s(H)$ and $row_s(H)$ denote the s-th column and row, respectively.

Given a set $S \subseteq X \times Y \subseteq \mathbb{R}^n \times \mathbb{R}^m$, the projection of the set S onto X is defined as $\text{Proj}_X(S) := \{x \in X \mid \exists y \in Y \text{ s.t. } (x,y) \in S\}$.

II. Problem Formulation

A FMS is an integrated, computer-controlled facility whose elements are automated material handling devices and numerically controlled machine tools that can simultaneously handle medium-sized volumes of pieces and parts to be processed. For control purposes, an FMS planimetry is fully described by the geometrical description of the accessible space where the AVs need to travel and the position of the Load/Unload (L/U) station and of the working machines $Ma = \{Ma_1, Ma_2, \dots, Ma_q\}$, see Fig. 1 for an example. The accessible region of the planar space is defined as:

$$\mathcal{O}_{free} := \bigcup_{k=1}^{n_c} \{x \in \mathbb{R}^n : p \in CO^k\} \tag{4}$$

where $p \in \mathbb{R}^2$ are the planar components of the state space $x \in \mathbb{R}^n$, while CO stands for corridor, i.e., the space connecting two locations in the FMS is formally defined by resorting to polyhedral arguments:

$$CO^k : \bigcap_{s=1}^{2} \{(W_s^k)^T p > (c^k)_s\} \tag{5}$$

The FMS is supposed to be composed of n_c corridors and each one of them described by intersection of hyperplanes.

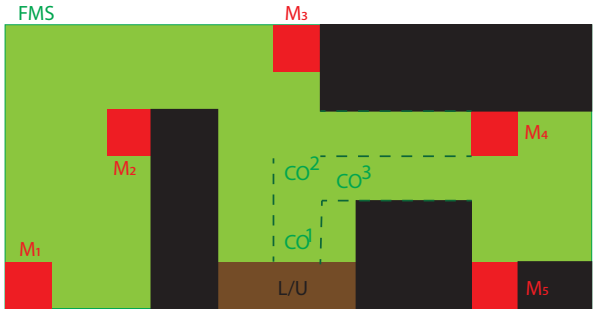

Fig. 1. A simple example of Smart Manufacturing Environment. Red squares represent machines, the brown one is the L/U station. $\mathcal{O}_{free}$ in green. Few corridors surrounded by dotted lines.

The production objective of the factory is the accomplishment of the task. The task is a collection of a finite number of jobs $\{J_1, \dots, J_h\}$, where a job is an ordered sequence of operations $op_i, i = 1, \dots, s$, carried out on dedicated machines $\{M_1, \dots, M_s\} \subseteq M$ [8].

In order to synthetize a model-based control strategy for the AVs in the FMS, a mathematical model is needed. Hereafter, the AV dynamics is described by means of discrete-time linear time invariant (LTI) models

$$x^i(t+1) = A^i x^i(t) + B^i u^i(t), \;\; i = 1, \dots, L, \tag{6}$$

subject to set-membership constraints:

$$\begin{aligned} x^i(t) \in \;& \mathcal{X}^i := \{x^i \in \mathbb{R}^{n_i} : {x^i}^T x^i \leq \bar{x}^2\}, \\ u^i(t) \in \;& \mathcal{U}^i := \{u^i \in \mathbb{R}^{m_i} : {u^i}^T u^i \leq \bar{u}^2\}, \; \forall t \geq 0, \end{aligned} \tag{7}$$

where $x^i \in \mathbb{R}^{n_i}$ is the state and $u^i \in \mathbb{R}^{m_i}$ the control input, with $\bar{x}, \bar{u} \in \mathbb{R}$ positive scalars. It is assumed that $x^i(t) = [p^i(t)^T, x_{np}^{i^T}]^T$ with $x_{np}^i \in \mathbb{R}^{n_i - 2}$ accounting for the non-spatial components and the structure of (6) includes integral actions in charge to ensure that any planar position is an equilibrium under zero velocity and $u^i(t) \equiv 0$. Moreover, the vehicle geometry is defined by resorting to a polyhedral description in the 2-D space:

$$AV^i : \begin{bmatrix} (H^i)_1^T \\ \vdots \\ (H^i)_{l_i}^T \end{bmatrix} p \leq \begin{bmatrix} (g^i)_1 \\ \vdots \\ (g^i)_{l_i} \end{bmatrix} \tag{8}$$

where ξ^i denotes the vehicle shell centroid, $(H^i)_j^T p \leq (g^i)_j$, $j = 1, \dots, l_i$ half-plane portions whose intersection is the vehicle shell and

$$d_{max} := \max_{j=1,\dots,l_i} dist\left(\xi^i, (H^i)_j^T p \leq (g^i)_j\right), \forall i \tag{9}$$

The on-board sensor module is in charge to detect obstacles within the vision radius $R > 0$ and a field of view of 360^o such that $R > R_{min}^c$, with R_{min}^c be the minimum curvature radius. Therefore, the detection region is:

$$\mathcal{B}(p^i(t) + d_{max}, R), \; \forall i = 1, \dots, L$$

Then, the problem of interest can be stated as follows.

Platoons in Flexible Manufacturing Systems (PL-FMS) - *Given a FMS $(M, L/U)$ geometrically defined by (5)-(4) and a group of AVs (6) organized as r platoons of L_j, $j = 1, \dots, r$, agents such that $\sum_{j=1}^{r} L_j = L$, determine a distributed state-feedback control policy*

$$\begin{aligned} u_j^1(t) &= g(x^1(t)) \\ u_j^i(t) &= g(x^i(t), x_j^{i-1}(t)), i = 2, \dots, L_j, j = 1, \dots, r, \end{aligned} \tag{10}$$

compatible with (4) and (7) such that an assigned task $\{J_1, \dots, J_h\}$, with associated job priorities $\{pr_1, \dots, pr_h\}$, $0 \leq pr_k \leq 1, pr_i \neq pr_j, \forall i \neq j$, is accomplished regardless any time-varying obstacle occurrence $\mathcal{O}_t := \{AV^k\}$. □

The **PL-MS** problem needs a formal definition of the scheduling of the prescribed task, the corresponding routing decisions scheme to plan the paths and the underlined control unit.

III. The Multi-Layer Control Architecture

The proposed approach integrates RL and Set-Theoretic MPC into a unified framework. Fig. 2 illustrates the overall architecture. Its core is the Routing/Decision unit. The input to this module is given by an external task scheduler in charge of converting the prescribed task $\{J_1, J_2, \dots, J_h\}$ into sequences of set-points for the AVs. This process exploits the FMS structure, i.e. , the involved machines M, the factory planimetry (5)-(4), and the available AVs team (6)-(8). The Routing/Control unit computes control-actions based on the given set-points and the actual states of the AVs. In particular,

- the routing decision $a^i(t)$, $i = 1, \dots, L$, are updated during on-line operations, on the basis of the scheduling

path $P_i^{goal} \in \mathbb{R}^{2\times\eta^i}$, $i = 1,\dots,h$, and the current state measurement $x^i(t)$, $i = 1,\dots,L$;
- the control action $u^i(t)$, $i = 1,\dots,L$, are computed in a distributed fashion and take into account the scheduling sequence $\{P_1^{goal},\dots,P_h^{goal}\}$, the list of job priorities $\{pr_1,\dots,pr_h\}$ and the routing decisions $\{a^1(t),\dots,a^L(t)\}$.

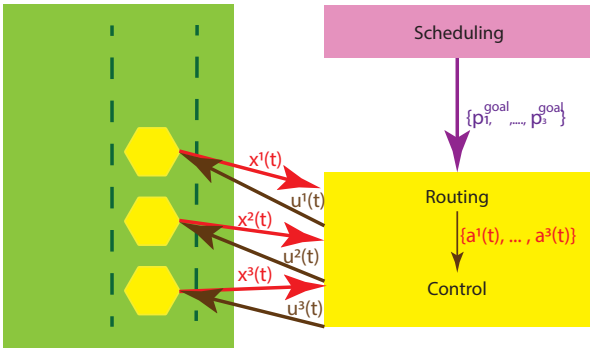

Fig. 2. Multi-layer control architecture

Essentially, the *modus operandi* can be summarized as follows. In the off-line phase, a centralized unit is in charge of computing a scheduling for the task, see [9]. This first layer provides a sequence of objective points to the AVs. Then, at each time instant t, the Routing layer computes the near-optimal path that the AVs need to follow in terms of routing decisions $\{a_1 \dots a_L\}$. For the sake of a formal description, the routing decision process is activated along any corridor once the platoon enters a prescribed corridor section, denoted as decision zone and defined as:

$$DZ^k: \; G^k p > b^k,\; k = 1,\dots,n_c \tag{11}$$

The translation of routing actions into set-points $z^i(t)$ for the control computation purposes is achieved by using a reference generator unit that receives in input the current routing decision $a^i(t)$ and computes the set-point according to the following rule

$$z^i(t) := \arg \max_{z^i \in CO^{a^i(t)}} \|p^i(t) - z^i\|^2 \tag{12}$$

These are exploited by the controllers to determine the control actions to be sent to the platoons. As highlighted in Fig. 2, Routing Decisions (and Control Actions) are online updated based on the actual data measured by the platoons.

The next two sections are devoted to the description of the proposed Routing/Control unit.

IV. A DEEP REINFORCEMENT LEARNING SCHEME FOR ROUTING DECISIONS

Following the lines proposed in [3], the routing decision layer is designed using a multi-agent deep Q-learning (DQL) framework [2]. This approach enables decentralized decision-making while leveraging learned policies for optimal route selection. The corresponding distributed DQL model is formulated as:

$$< V, \Sigma, \Lambda, \Phi > \tag{13}$$

This model consists of a set of agents $V := \{AV_j^i\}_{i=1}^{L_j}$ introduced to describe the behavior of vehicles moving on the FMS. The second term in 13, corresponds to the set of **DQL state** - $\Sigma_j := \{(\sigma_j^i)_R(t)\}_{i=1}^{L_j}$. Specifically, σ_j^i is the edge where the vehicle AV_j^i is located at the time instant t. The finite set Λ collects the admissible vehicle actions, encoded as integer numbers. $\Psi : \Sigma \times \Lambda \times \Sigma \to \mathbb{R}$ is named **Global reward** and takes care of vehicle congestion occurrences over the FMS:

$$\Psi\left(\bigcup_k \sigma_R^k(t_{dec} + \Delta t_{dec}(t))\right)$$

where t_{dec} refers to the time instant when a routing decision must be taken because enters the decision zone:

$$\Delta t_{dec}(t) := t - t_{dec} with p_j^i(t_{dec}) \in CO^{k_1} \cap CO^{k_2}, j_1 \neq j_2$$

The global reward is received by all agents AV_j^i in linear combination with some heuristics, defining the local reward:

$$\begin{aligned}\psi_j^i(t_{dec} + \Delta t_{dec}(t)) = \; & \Psi\left(\bigcup_k (\sigma_j^k)_R(t_{dec} + \Delta t_{dec}(t))\right) + \\ & (\psi_j^i)_{goal}(t_{dec} + \Delta t_{dec}(t)) + \\ & (\psi_j^i)_{\Delta w_j^i}(t_{dec} + \Delta t_{dec}(t))\end{aligned} \tag{14}$$

where:

- $(\psi_j^i)_{goal}(t_{dec} + \Delta t_{dec}(t))$ is in charge of considering the actual current longitudinal vehicle's speed divided by the distance from the target;
- $(\psi_j^i)_{\Delta w_j^i}$ is defined as

$$\begin{aligned}&(\psi_j^i)_{\Delta w_j^i}(t_{dec} + \Delta t_{dec}(t)) = \\ &-\Delta w_j^i(t_{dec} + \Delta t_{dec}(t)) + (f_j^i)_{loop}(t_{dec} + \Delta t_{dec}(t))\end{aligned} \tag{15}$$

with Δw_j^i the total waiting time spend by the vehicle AV_j^i to reach $t_{dec} + \Delta t_{dec}(t)$ and $(f_j^i)_{loop}$ penalizes possible loops along the traveling.

Notice that the reward (14) is computed only when the generic vehicle AV^i has effectively performed the imposed action.

For learning purposes the FMS environment is associated to a connected graph $G = (V, \mathcal{E})$, with each edge representing a corridor CO^k and where two adjacent edges correspond to two close corridors.

The agents learn the **State-action value function:**

$$\begin{aligned}&Q_j^i((\sigma_j^i)_R(t), (a_j^i)_R(t); \theta_j^i) := \\ &E\left[\sum_{\nu=1}^{\infty} \gamma^{(\nu-1)}\, \phi^i(t + \nu \Delta t_{dec}(t))\right]\end{aligned} \tag{16}$$

where $(a_j^i)_R(t) \in \Lambda$, θ_j^i is the NN weighting vector of the vehicle AV_j^i and $\gamma \in (0,1)$ the discount factor. The **Policy** is epsilon-greedy [11] and given as

$$(a_j^i)_R = \begin{cases} rand(\Lambda), \text{with probability } \epsilon \\ \arg\max_{a\in\Lambda} Q_j^i((\sigma_j^i)_R(t_{dec}), a; \theta_j^i), \text{otherwise} \end{cases} \quad (17)$$

All these elements have been included in the proposed Routing-Control unit. It makes use of the following computable algorithm:

D-DQL-RL Algorithm (Agent AV_j^i)

Initialize $Q_j^i((\sigma_j^i)_R, a_R; \theta_j^i)$

For *each episode*

1: **Read** $(\sigma_j^i)_R(0)$;

At *each* t_{dec}

1: **Select** by (17) the action $(a_j^i)_R$ to be executed;
2: **Read** the DQL state $(\sigma_j^i)_R(t_{dec} + \Delta t_{dec}(t))$;
3: **Compute** the reward $\psi_j^i(t_{dec} + \Delta t_{dec}(t))$;
4: **Update** θ_j^i;

V. The proposed set-theoretic control strategy

The last layer of the proposed control architecture computes feasible command inputs to accomplish the prescribed task by following the path prescribed by the routing layer. It must satisfy the platoon formation, fulfill input/state constraints (7) and avoid unexpected obstacles along the path. This is done by carefully formalizing off-line and on-line phases of the set-theoretic based control strategy. First of all, according to the scheduling path $P_i^{goal} \in \mathbb{R}^{2\times\eta^i}$, $i = 1,\dots,L$, the AVs team (6)-(8) is partitioned into $r \leq L$ platoons $\mathbf{PL}_j$. Then, the following operation are performed:

- **Off-line phase -** For each corridor CO^k, a corresponding state trajectory tube satisfying constraints (4) and (7) is precomputed for each leader-follower configuration $\mathbf{PL}_j$. The union of all of these regions is the Domain of Attraction (DoA^j) of the algorithm. All of these regions are systematically stored in a designated repository for on-demand retrieval during real-time operations. Notice that, since the operations take place in structured indoor environments, it is assumed, without loss of generality, that the required data can be efficiently and instantaneously accessed by AVs, for instance, via an *ad-hoc* Wi-Fi network.
- **On-line phase -** Each AV is controlled by a distributed model predictive control (DMPC) unit, synthesized according to the prescribed platoon formation. At each time step t, the μ platoons operate concurrently to achieve their assigned goals P_i^{goal}. The control actions $u^i(t), \forall i$ are computed based on the routing decisions $a^i(t)$. If two platoons $\mathbf{PL}_a = \{AV_a^1, AV_a^2, \dots\}$ and $\mathbf{PL}_b = \{AV_a^1, AV_a^2, \dots\}$ travel in the same corridor CO^k in opposite directions they perceive each other based on the condition:

$$\mathcal{B}(p_a^1(t) + d_{max}, R) \cap \mathcal{B}(p_b^1(t) + d_{max}, R) \neq \varnothing \quad (18)$$

In this case, they exchange a message about their associated priorities pr_a, pr_b to be compared. The lower-priority platoon, w.l.g. $pr_b < pr_a$, acts as an obstacle to be passed by the other platoon.

A. Offline phase

The fundamental theoretical result behind the computation of the sets covering is Preposition 8 in [6]. There, it has been proved that given two overlapped RPI regions, if the state of the system belongs to one of them, the problem of computing a control action to bring the state to the other one is always feasible. Therefore, sequence of overlapped RPI sets have been computed for each leader and for each corridor, through the following optimization problem:

$$\min_{(K^1)_k^j, (\mathcal{T}_0^1)_k^j} J_\infty^1(t) \text{ s.t} \quad (19)$$

$$(A^1 + B^1(K^1)_k^j)(\mathcal{T}_0^1)_k^j \subset (\mathcal{T}_0^1)_k^j \subseteq (\mathcal{X}_c^1)^j := \{\mathcal{X}_j^i\}_{i=1}^{L_j} \cap CO^k \quad (20)$$

$$(K^1)_k^j (\mathcal{T}_0^1)_k^j \subset \mathcal{U}^1, \quad (21)$$

$$(\bar{x}^1)_k^j \in (\mathcal{T}_0^1)_{k-1}^j \cap (\mathcal{T}_0^1)_k^j \quad (22)$$

Then, regions for each follower are computed through a similar optimization problem 19-22, adding the formation constraint:

$$d_{min} \leq dist(x^i, (\mathcal{T}_0^{i-1})_k^j) \leq d_{max}, \forall x^i \in (\mathcal{T}_0^i)_k^j \quad (23)$$

where $(\bar{x}^i)_k^j, i = 1,\dots,L_k$, $k = 1,\dots r$, $j = 1,\dots n_c$ is the equilibrium selected at each $k-th$ iteration as follows

$$\begin{aligned} (\bar{x}^i)_k^j := \quad & \arg \max_{x\in(\mathcal{T}_0^i)_{k-1}^j} \|(\bar{x}^i)_{k-1}^j - x^i\|_2 \\ & \text{s.t. } \exists \bar{u}^i \in \mathcal{U}^i \,,\; x^i = (I_n - A^i)^{-1} B^i \bar{u}^i \end{aligned} \quad (24)$$

and

$$\begin{aligned} J_\infty^i(t) := & \sum_{\tau=0}^{\infty} \left[\|x^i(t+\tau|t) - x_{fin}^i\|_{R_x^i}^2 + \|u^i(t+\tau|t)\|_{R_u^i}^2 \right], \\ & i = 1,\dots,N, \end{aligned} \quad (25)$$

with $R_x^i > 0$ and $R_u^i \geq 0$ symmetric state and input weighting matrices, respectively.

B. Online phase

During the online phase, for each vehicle, depending on which region its actual state belongs, the corresponding feedback gain is applied, in a Receding Horizon Control fashion.

More complex to deal with is the case of two platoons travel in opposite directions along the same corridor because an anticollision procedure is necessary. Essentially, as depicted in Fig. 3, the two platoons exchange a message about their respective priorities. Therefore, the lower-priority platoon stops and acts as an obstacle. Meanwhile, the higher-priority platoon synthesizes a tube of overlapped RPI regions (19)-(23). This new tube enables the higher-priority platoon to safely pass the other.

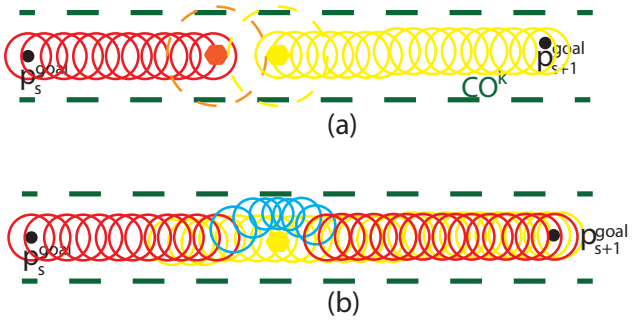

Fig. 3. Avoiding low-priority platoon. (a) The two platoons are traveling in opposite directions. Dotted lines represent vision radius. (b) The tubes used by the two platoons, The RPI regions synthesized online are highlighted in cyan.

C. Distributed DRL-MPC algorithm

The developments of the previous sections have been collected to write the following computable algorithm.

DRL-MPC Algorithm - Vehicle $i-th$ **,** $AV_i \in \mathbf{PL}_j$

Input: d_{min}, d_{max}, x_{in}^i, x_{fin}^i;

Initialization-

1: **Compute** $\{(\mathcal{T}_0^1)_k^j, (K^1)_k^j\}$ by solving (19)-(22);
2: **Compute** $\{(\mathcal{T}_0^i)_k^j, (K^i)_k^j\}$ complying with (19)-(23);
3: **Derive** the equilibria by (24);
4: $k \leftarrow k+1$; goto **Step 1**;

On-line phase -

1: **Receive** $x^i(t)$;
2: **if** for some $\mathbf{PL}\bar{j}$ (18) holds with j and $\bar{j}$ in place of a and b **then** go to Step 4 ▷ a collision may happen
3: **Determine** the routing decision $z^i(t)$;
4: **if** $pr_j > pr_{\bar{j}}$ **then Block** the platoon $\mathbf{PL}_{\bar{j}}$
5: **if** $i=1$ **then compute** the PI region $(\mathcal{E}_{j,i}^k)$ by solving the **DMPC**-Leader optimization (19)-(22);
6: **else**
solve the **DMPC**-Follower optimization (19)-(23);
7: **end if**
8: **Compute** $u_j^{i^*}(t) = \left(K_{j,i}^k(t)\right) x_j^i(t)$
9: **else**
Block the platoon $\mathbf{PL}_j$: $u^i = 0$
10: **Repeat** Steps 5-8 with $\bar{j}$ in place of j;
11: **end if**
12: **else**
13: **Compute** $u_j^{i^*}(t) = \left(K_{j,i}^k(t)\right) x_j^i(t)$
14: **end if**
15: **Apply** $u^{i^*}(t|t)$;
16: $t \leftarrow t+1$ and goto **Step 1**;

VI. Simulations

In this section, the effectiveness of the proposed **DRL-MPC** algorithm is evaluated by means of a simulation campaign, whose setup is implemented within the MATLAB environment. For the subsequent developments, autonomous vehicles are described by the following 4-state (lateral, longitudinal positions and velocities) 2-inputs (lateral and longitudinal accelerations) double-double integrator discretized LTI model (6) whose dynamic and control input matrices are as follows:

$$A^i = \begin{bmatrix} I_2 & T_s I_2 \\ 0_2 & I_2 \end{bmatrix}, \; B^i = \begin{bmatrix} \frac{(T_s)^2 I_2}{2} \\ T_s I_2 \end{bmatrix},$$

with $T_s = 1sec$ and subject to the saturation constraint $||(u^i(t))|| \leq 0.5, \forall t \geq 0, \forall i$.

Under the platoon configuration, the geometrical displacement between two consecutive vehicles is $d_{min} = 0.5[m]$ and $d_{max} = 1.5[m]$.

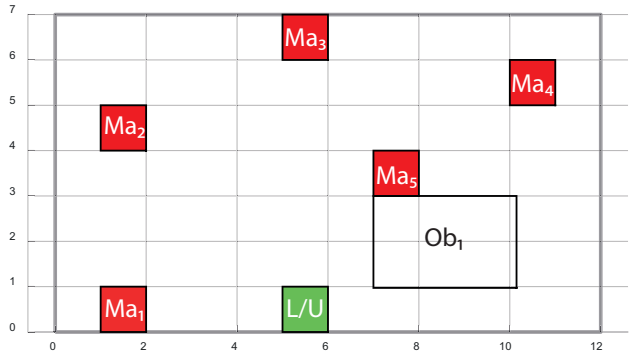

Fig. 4. FMS planimetry: red squares represent machines. The green square is the L/U station. The white square named Ob_1 is a region forbidden to the AVs.

The goal of the simulation campaign is to control three autonomous vehicles organized into two platoons using the proposed **DRL-MPC** algorithm. The simulation scenario is the FMS given in Fig. 4, defining an area of $84\,[m^2]$ and composed of the L/U station and four machines.

The training process of the Q-function (16) has been addressed within the MATLAB environment by using Reinforcement Learning Toolbox built-in routines. Fig. 5 shows the average reward (defined as the moving average of the episode reward - the sum of all the received rewards during each episode -) and the expected reward. This plot proves how proper routing decisions are selected according to the training evolution: the average reward (blue line) settles down to an high performance value, while the expected reward (orange line) asymptotically converges to it.

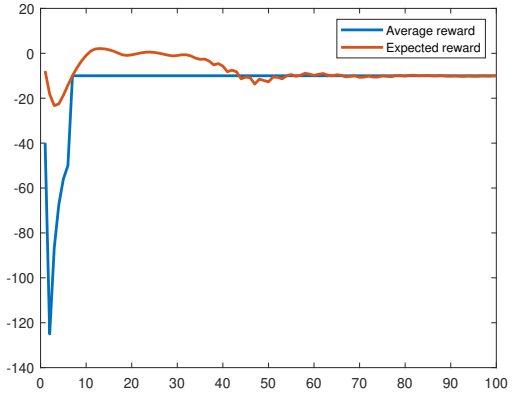

Fig. 5. DRL training: Average rewards *versus* Expected reward

Fig. 6 shows time evolution of the trajectories of the two platoons to assist logistics operations. Fig. 7 illustrates the same planar trajectories on the considered FMS. In details, $\mathbf{PL_1}$ goes from the L/U station to machine 4, then to machine 2, machine 1 and back to the L/U station. Paralleling, $\mathbf{PL_2}$ goes from the L/U station to machine 3, then to machine 2, machine 5, and back to the L/U station.

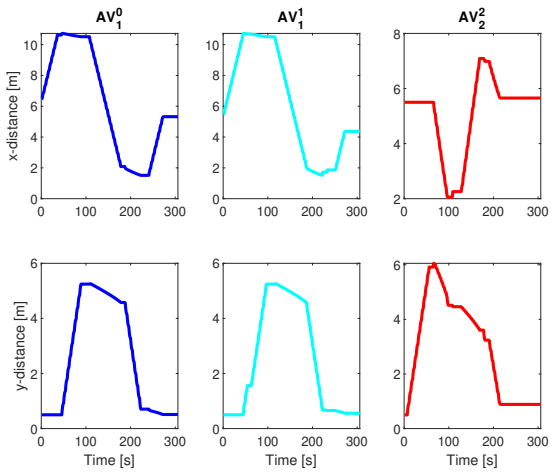

Fig. 6. Time evolution of x and y coordinates of the three AVs

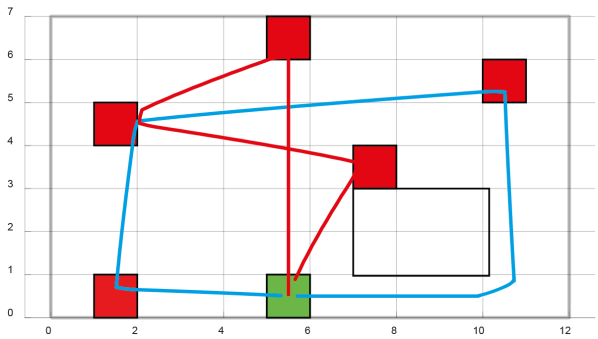

Fig. 7. Trajectories on the FMS. The blue line is the trajectory of $\mathbf{PL_1}$, the red one is the trajectory of $\mathbf{PL_2}$

Fig. 8 shows the fulfillment of the constraints during the online phase. Specifically, the upper part illustrates the constraint on the control inputs while the lower part presents the formation constraint relative to the two-vehicles platoon.

VII. Conclusions

In this paper, a novel distributed control architecture for logistics operations in flexible manufacturing systems has been designed. The core of the proposed solution is based on two methodologies: deep reinforcement learning and distributed receding horizon control. The core concept involves dividing the available fleet of vehicles into platoons to minimize task completion time and enhance efficiency. The proposed distributed control framework incorporates coordination and collision avoidance capabilities, which are integrated based on real-time updates from the scheduling unit, continuously refined by the reinforcement learning algorithm.

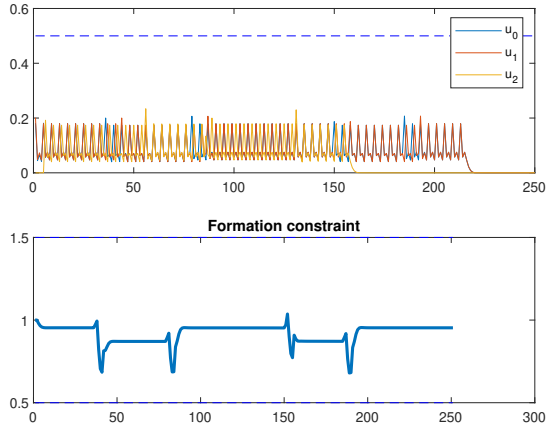

Fig. 8. Fulfillment of the constraints during the online phase

Acknowledgment

This work was partially supported by project SERICS (PE00000014) under the MUR National Recovery and Resilience Plan funded by the European Union - NextGenerationEU.

References

[1] F. Blanchini and S. Miani,"Set-Theoretic Methods in Control", *Birkäuser*, Boston, 2008.

[2] L. Canese et al.,"Multi-agent reinforcement learning: A review of challenges and applications", *Applied Sciences*, Vol. 11, No. 11, 2021.

[3] L. D'Alfonso, F. Giannini, G Franzè, G. Fedele, F. Pupo, G. Fortino,"Autonomous Vehicle Platoons In Urban Road Networks: A Joint Distributed Reinforcement Learning and Model Predictive Control Approach", *IEEE/CAA J. of Auto. Sinica*, Vol. 11, No. 1, pp. 1-16, 2024.

[4] C. Dey and S. K. Sen, "Industrial automation technologies," *CRC Press*, 2020.

[5] G. Franzè, D. Famularo and F. Tedesco Fortino, "Receding horizon control for constrained networked systems subject to data-losses", *Proceedings of the IEEE Conference on Decision and Control*, Pp. 5260 - 5265, 2011.

[6] G. Franzé and W.Lucia,"A Receding Horizon Control Strategy for Autonomous Vehicles in Dynamic Environments", *IEEE Transaction on Control Systems Technology*, Vol. 24, No. 2, 2016.

[7] F. Giannini, G. Franzè, F. Pupo and G. Fortino, "A sustainable multi-agent routing algorithm for vehicle platoons in urban networks", *IEEE Transactions on Intelligent Transportation Systems*, 2023.

[8] J. Hurink and K. Sigrid,"Tabu Search Algorithms for Job-shop Problems with a Single Transport Robot. Logistics: From Theory to Application", *European J. of Operational Research*, Vol. 162, N. 1, pp. 99-111, 2005.

[9] Hu, Liang, et al,"Petri-net-based dynamic scheduling of flexible manufacturing system via deep reinforcement learning with graph convolutional network", *Journal of Manufacturing Systems*, N. 55, Pp. 1-14, 2020.

[10] R. Sell et al, "Integration of autonomous vehicles and Industry 4.0," *Proceedings of the Estonian Academy of Sciences*, Vol. 68, N. 4, Pp. 389-394, 2019.

[11] R. Norvig and S. Russel, "Artificial Intelligence. A modern approach," *Pearson* 2021.

[12] R. Sutton and A. Barto, "Reinforcement learning: An introduction," *MIT press*, 2018.

[13] L. Willems, "Understanding the impacts of autonomous vehicles in logistics," *The Digital Transformation of Logistics: Demystifying Impacts of the Fourth Industrial Revolution*, Pp. 113-127, 2021.

[14] A. Yadav and S. C. Jayswal, "Modelling of flexible manufacturing system: a review," *International Journal of Production Research*, Vol. 56, No. 7, Pp. 2464-2487, 2018.