2020 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)
October 25-29, 2020, Las Vegas, NV, USA (Virtual)

Predicting the human behaviour in human-robot co-assemblies: an approach based on suffix trees

Andrea Casalino[1], Nicola Massarenti[1], Andrea Maria Zanchettin[1] and Paolo Rocco[1]

Abstract— **Prediction of the human behaviour is essential for allowing an efficient human-robot collaboration. This was confirmed recently showing how scheduling approaches can significantly increase the productivity of a robotic cell by planning the robotic actions in a way as much as possible compliant with the human predicted behaviour. This work proposes an innovative approach for human activity prediction, exploiting both a-priori information and knowledge revealed during operation. The resulting approach is proved to achieve good performance through both off-line simulated sequences and in a realistic co-assembly involving a human operator and a dual arm collaborative robot.**

I. Introduction

Collaborative robotics is emerging as an important research line within the robotics community. Much attention was given to industrial contexts, where humans and robots collaborate to accomplish structured tasks, which can be difficult to fully automatize by exploiting only robots. Although many different applications were proposed, few key abilities were proved to allow for an efficient human-robot interaction among which we can find the capability to predict the human behaviour.

When considering a physical interaction, typical approaches predict the human behaviour in terms of the future human motion, in order to enhance impedance controls. Such a prediction is done in [1] by exploiting a machine learning approach based on radial basis neural networks, able to forecast the human motion according to the force applied at the end effector. A similar approach is exploited in [2], where neural networks are exploited for improving an adaptive impedance control scheme.

When dealing with co-assemblies, humans and robots have to alternate and synchronize to finalize a set of products. In such contexts, the intention prediction is considered at a task level, i.e. predicting the sequence of future human actions according to the past ones. Cyber-physical systems [3] can be exploited to optimally control the robotic actions, according to a digital model describing the plant to supervise and in particular the precedence constraints among the actions assigned to both humans and robots. The actions done by agents are recorded and notified to the digital model for updating the state of the system. Recent works have demonstrated the benefits of predicting the human behaviour [4], [5] for improving the scheduling capabilities of cyber-physical systems.

The predictive models can be also adopted for determining the waiting time to see again a certain specific action assigned to the human [6]. In this context, the human activity recognition becomes of paramount importance. We can assume humans in the robotic cells as constantly monitored by surveillance systems, able to track the human motion. Then, algorithms like [7] or [8] are able to solve the action recognition problem by analyzing the trajectory of some specific points of the human upper body, detecting also the starting and ending time instants of the human actions. In this way, an artificial intelligence is able to keep track of the operations performed during time by the human. Finally, algorithms performing time series prediction can be exploited to determine the future actions the human will undertake.

The time series prediction problem was already addressed in the literature. In [9] the concept of prediction by partial matching was introduced. It describes the time series evolution by means of transition probabilities. In [10] Support Vector Machines are used for hierarchical multi-label prediction of gene functions. Li et al. [11] proposed a Variable Order Markov model approach, able to represent both high and low temporal correlations. Gueniche et. al proposed another approach, based on a tree of suffixes [12], [13]. The approach proposed in this paper takes inspiration from these works, extending them with the adoption of a probabilistic perspective. In [14], Recurrent Neural networks are exploited, while another kind of network is exploited in [15] for predicting stock market prices.

When considering the possible actions assigned to a human in a robotic cell, it is common to have affine actions, for which the probability to be executed in sequence is high. Classical probabilistic models (Markov chains, Bayesian networks, etc.) are trained in a data-driven way, trying to highlight the temporal dependencies among the actions. The relationship between affine actions might be included as an a-priori knowledge, which must be however modelled as a probability distribution of some kind that must be included in the model with severe modifications. On the opposite, the aim of this work is to propose an innovative algorithm that considers both the actions relationships as well as their temporal dependencies in a unique time varying model, which is the main innovation of our approach.

II. Predict the human actions

We assume the actions assigned to the human in an assembly task as contained in a finite set $\mathcal{A} = \{a_1, \cdots, a_m\}$. Every element in $\mathcal{A}$ is an elementary action which requires one operator's hand to enter in a certain area of the workspace, taking tools or parts to assemble. The sequence of actions

[1] The authors are with Politecnico di Milano, Dipartimento di Elettronica, Informazione e Bioingegneria, Piazza L. Da Vinci 32, 20133, Milano, Italy (e-mail: name.surname@polimi.it).

performed during time can be described by a time series $X = x_1 \rhd x_2 \rhd \cdots \rhd x_n$, with $x_{1,2,\cdots,n} \in \mathcal{A}$ [1]. Timestamps $t_{1,2,\cdots,n}$ of the elements in X refer to instants at which the operator begins the corresponding action. In order to optimally plan the operations assigned to robots, the future values of X must be predicted. This is possible by adopting the model described in this Section, which is made of two main parts: the first one models the logical sequence of operations (Section II-A), while the second one accounts for the temporal durations (Section II-B).

A. Predicting the sequence of human actions

The process governing the time series X is assumed to be stochastic. The conditional probability to see a certain action $a \in \mathcal{A}$ as a realization for x_n must be characterized. Such distribution can be built by taking into account σ preceding actions, i.e. considering the sub-series $X_n^\sigma = x_{n-\sigma} \rhd \cdots \rhd x_{n-1}$. In this work we propose to use a Gibbs distribution to model the following conditional distribution:

$$\mathbb{P}(x_n = a | X_n^\sigma) = \frac{\Psi(X_n^\sigma, a, t)}{\sum_{\tilde{a} \in \mathcal{A}} \Psi(X_n^\sigma, \tilde{a}, t)} = \frac{\Psi(X_n^\sigma, a, t)}{Z\big(ST(t)\big)} \quad (1)$$

σ is also referred to be the order of the predictive model. The factors characterizing Ψ are all exponentials:

$$\Psi = \Psi_v\big(X_n^\sigma, a, t\big) \cdot \prod_{i=1}^{N_C} \Psi_{ci}\big(X_n^\sigma, a\big) \quad (2)$$

$$\Psi_v = exp\Big(w_0 \Phi_v\big(X_n^\sigma, a, t\big)\Big) \quad (3)$$

$$\Psi_{Ci} = exp\Big(w_i \Phi_{Ci}\big(X_n^\sigma, a\big)\Big) \quad (4)$$

The adoption of a Gibbs distribution made by exponential linear factors was done to have a model whose logarithmic likelihood is easy to differentiate, making the training computationally affordable (see Section II-C). Φ_v is a piecewise time varying function, depending on the definition of a suffix tree, see Section II-A.1. A suffix tree is a dynamic data structure storing all the information acquired during time about the time series to predict. On the opposite, functions $\Phi_{c1,\cdots,C}$ remain invariant and are assumed as given. They model an a-priori knowledge to be used in the prediction process, see Section II-A.2. Equation (1) is used for the single step prediction. Then, by recursively propagating it, the probability of a sequence $x_n \rhd x_{n+1} \rhd \cdots \rhd x_{n+L}$, conditioned to X_n^σ can be also evaluated. The computations for $x_n \rhd x_{n+1}$ will be detailed, then it is easy to extend the reasoning to the general case:

$$\mathbb{P}(x_n = a_0, x_{n+1} = a_1 | X_n^\sigma) = \mathbb{P}(x_n = a_0 | X_n^\sigma) \cdot \mathbb{P}(x_{n+1} = a_1 | x_{n-\sigma+1} \rhd \cdots \rhd x_{n-1} \rhd a_0) \quad (5)$$

The two factors in the above equation are computable by making use of equation (1). The conditional probability of x_{n+L} w.r.t X_n^σ, regardless the intermediate values $x_{n,\cdots,n+L-1}$ could be computed with the following summation:

$$\mathbb{P}(x_{n+L} = a_L | X_n^\sigma) = \sum_{\tilde{a}_{0,\cdots,L-1} \in \mathcal{A} \times \cdots \times \mathcal{A}} \mathbb{P}(x_n = \tilde{a}_0, \cdots, x_{n+L} = a_L | X_n^\sigma) \quad (6)$$

Each terms in the above summation can be evaluated by making use of equation (5).

The following Sections show how to build the two kind of factors contained in equation (2), which are the ones required for computing the one-step predicting distribution of equation (1).

1) Definition of the suffix tree: A suffix tree (ST) is a time varying structure: every time a new value x_{n+1} is available, the tree is updated. A ST describes in a compact way the information contained in the sequence $x_0 \rhd \cdots \rhd x_n$. To every node, excluding the root, an action $a \in \mathcal{A}$ is assigned. The path connecting the root with the i^{th} leaf, also called branch, is denoted as B^j and is an ordered sequence of actions $x_{Bj1} \rhd x_{Bj2} \rhd \cdots$. The population of all the branches of the tree contains all the observed sub-sequences in X, up to step k.

When considering a particular model order σ, each branch in the tree will have a length equal to $\sigma + 1$. A set of tokens $\Gamma^j = \{\gamma_1^j, \gamma_2^j, \cdots\}$ is assigned to the j^{th} leaf, whose meaning will be clear later.

Every ST is initialized with the presence of the sole root. The sequence $x_1 \rhd \cdots \rhd x_{\sigma+1}$ is inserted as first branch B^1 at step $\sigma + 1$, i.e. after observing the first $\sigma + 1$ values of X. At the same step, set Γ^1 is initialized with a single token $\gamma_1^1 = \sigma + 1$. Then, at the generic step n the ST is updated in this way:

- Case a): $X_n^\sigma \rhd x_n$ is already contained in the ST, i.e. there exists a branch $B^j = X_n^\sigma \rhd x_n$. In this case, a token equal to n is added to Γ^j, i.e. $\Gamma^j = \Gamma^j \cup n$.
- Case b): $X_n^\sigma \rhd x_n$ is not present in the ST. In this circumstance, a new branch $B^m = X_n^\sigma \rhd x_n$ is inserted in the tree, whose corresponding set Γ^m is initialized with the value n.

Fig. 1 reports some examples. Function Φ_v, equation (3), depends on an ST structure. Prior to define Φ_v, the operators $\mathcal{I}[\cdot]$ and $\mathcal{O}[\cdot]$ must be introduced. $\mathcal{I}$ describes the kind of actions contained in a sequence $Y = y_1 \rhd y_2 \cdots \rhd y_s$, regardless their order, and is defined as follows:

$$\mathcal{I}[Y]_{a_i} = \sum_{j=1}^{s} L(y_j)_{a_i} \quad a_i \in \mathcal{A} \quad (7)$$

where the indicator function L is here defined:

$$L(y_j)_{a_i} = \begin{cases} 1 & \text{if } y_j = a_i \\ 0 & \text{otherwise} \end{cases} \quad (8)$$

On the opposite, $\mathcal{O}$ aims at describing the way actions are disposed in a sequence and is defined in this way:

$$\mathcal{O}[Y]_{a_i}^K = \begin{cases} 0 \text{ if} & \mathcal{I}[Y]_{a_i} < K \\ \text{minimum } k \;\; s.t. & \mathcal{I}[Y^k]_{a_i} = K \end{cases} \quad (9)$$

[1] The notation $x_a \rhd x_b$ is used for expressing the fact that x_b was done immediately after x_a.

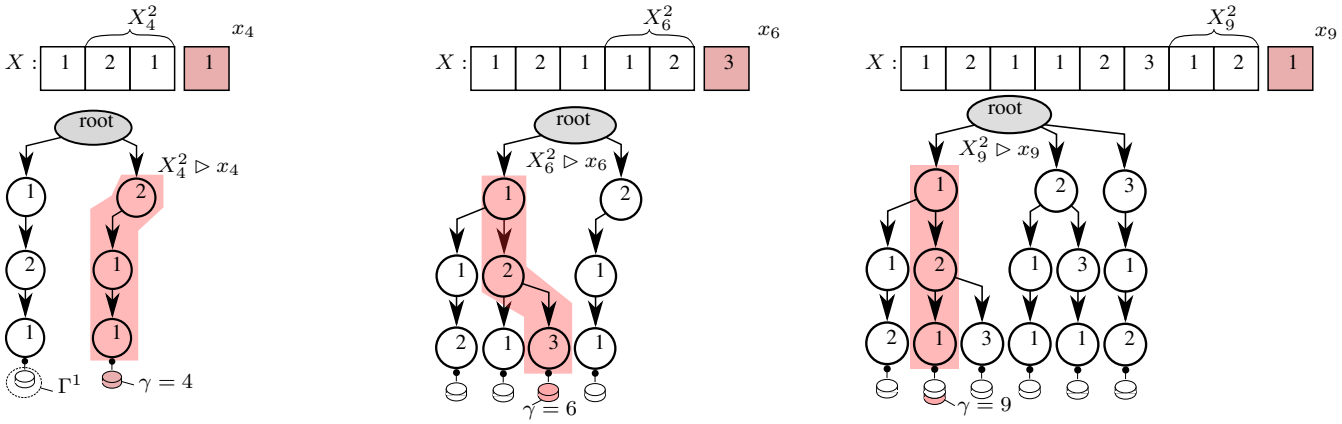

Fig. 1: Examples of suffix tree updates. The structure of the tree after the update is reported for each example. The token sets Γ associated to the leaves are indicated in the lower part of the pictures containing the trees.

$Y = \{2, 1, 3, 3, 2\}$

$$\left[\begin{array}{c} \mathcal{I}[Y]_1 = 1 \\ \hline \mathcal{I}[Y]_2 = 2 \\ \hline \mathcal{I}[Y]_3 = 2 \end{array}\right] \qquad \left[\begin{array}{c|c} \mathcal{O}[Y]_1^1 = 2 & \mathcal{O}[Y]_1^2 = 0 \\ \hline \mathcal{O}[Y]_2^1 = 1 & \mathcal{O}[Y]_2^2 = 5 \\ \hline \mathcal{O}[Y]_3^1 = 3 & \mathcal{O}[Y]_3^2 = 4 \end{array}\right]$$

TABLE I: Results obtained when applying operators $\mathcal{I}$ and $\mathcal{O}$ on the series Y reported at the top.

with Y^k indicating the sub portion of Y truncated at step k, i.e. $Y = y_1 \triangleright \cdots \triangleright y_k$. Refer to the example reported in Table I. Two possible distances, d_I and d_O can express the similarity existing between two sequences X and Y. They are defined according to the two previously introduced operators:

$$d_I(X, Y) = \sum_{i=1}^{m=|\mathcal{A}|} \left| \mathcal{I}[X]_i - \mathcal{I}[Y]_i \right| \tag{10}$$

$$d_O(X, Y) = \sum_{j=1}^{J} \sum_{i=1}^{m=|\mathcal{A}|} \left| \mathcal{O}[X]_i^j - \mathcal{O}[Y]_i^j \right| \tag{11}$$

with J equal to the length of the X (or Y). The domain of Φ_v is divided into three disjoint regions $\mathcal{D}_I(ST), \mathcal{D}_{II}(ST), \mathcal{D}_{III}(ST)$ (refer to equation (3)):

$$\Phi_v(X_n^\sigma, a|ST) = \begin{cases} \Phi_{vI} & \text{if } \{X_n^\sigma, a\} \in \mathcal{D}_1 \\ \Phi_{vII} & \text{if } \{X_n^\sigma, a\} \in \mathcal{D}_2 \\ \Phi_{vIII} & \text{if } \{X_n^\sigma, a\} \in \mathcal{D}_3 \end{cases} \tag{12}$$

Set $\mathcal{D}_1$ contains those sequence already existing in the ST. More formally:

$$\mathcal{D}_1 = \{X_n^\sigma, a \mid \exists B^j \in ST \ \ s.t. \ \ B^j = X_n^\sigma \triangleright a\} \tag{13}$$

Then, the complement of $\mathcal{D}_1$, is divided into two parts: the first one contains all those sequences for which in the ST there exists at least one branch having the same actions (with a different order) while the second one contains all the remaining ones. Assume operator $\mathcal{V}$ defined in the following way:

$$\mathcal{V}[X, ST] = \{B^j \in ST \mid d_I(B^j, X) = 0\} \tag{14}$$

Then, it holds that:

$$\mathcal{D}_2 = \{X_n^\sigma, a \mid \mathcal{V}[X_n^\sigma \triangleright a, ST] \neq \emptyset\} \tag{15}$$

$$\mathcal{D}_3 = \{X_n^\sigma, a \mid \mathcal{V}[X_n^\sigma \triangleright a, ST] = \emptyset\} \tag{16}$$

We are now in position to discuss the definition of Φ_{vI}, Φ_{vII} and Φ_{vIII}. To this purpose, the activation function f_{act}^Γ must be introduced:

$$f_{act}^\Gamma(n) = \sum_{\gamma \in \Gamma} exp\Big(-\alpha(n - \gamma) \Big) \tag{17}$$

The parameter α is determined in order to verify that $f_{act}^\Gamma(N) \cong 0$ for $N > \frac{5}{\alpha}$, with N a desired forgetting time that can be tuned considering the lengths of the human assembly sequences. Φ_{vI} is defined as follows:

$$\Phi_{vI} = f_{act}^{\Gamma^j}(n) \tag{18}$$

The Γ^j in the above equation is the one related to branch $B^j = X_n^\sigma \triangleright a$. Therefore, the aim of tokens is to activate more those sequences recently seen. However, since a summation is present in equation (17) a high activation value is provided also by those sequence seen many times. Φ_{vII} is defined in this way:

$$\Phi_{vII} = \frac{1}{|S|} \Big(\sum_{B^i \in S} \frac{1}{\beta} f_{act}^{\Gamma^i}(n) \Big) \tag{19}$$

$$where \ \ S = \mathcal{V}[X_n^\sigma \triangleright a, ST] \tag{20}$$

$$\beta = d_O\big(B^i, X_n^\sigma \triangleright a\big) \tag{21}$$

Finally, the definition of Φ_{vIII} is as follows:

$$\Phi_{vIII} = \frac{1}{|ST|} \Big(\sum_{B^i \in ST} \frac{1}{\delta} f_{act}^{\Gamma^i}(n) \Big) \tag{22}$$

$$where \ \ \delta = d_I\big(B^i, X_n^\sigma \triangleright a\big) + d_O\big(B^i, X_n^\sigma \triangleright a\big) \tag{23}$$

2) Handling the prior knowledge of the process: As humans, we are easily able to make predictions by exploiting contextual information. For instance, when someone takes a screwdriver, we naturally think that a subsequent action will involve screws. Similarly, when we see an operator gluing a surface, we guess that in the near future something will be attached. For this reason, we developed our method so as to manage some prior information regarding the process to predict. More formally, the generic activation function Φ_{Ci} (equation (1)), expresses the circumstance that a subset of actions $\mathcal{C}_i \subseteq \mathcal{A}$ are affine. The evaluation of Φ_{Ci}, is done as follows:

$$\Phi_{Ci}(X_n^\sigma, a) = \sum_{j=1}^{\sigma} L_{Ci}(x_{n-j}, a) \cdot exp\Big(-\alpha \cdot j \Big)$$

$$where \quad L_{Ci}(x,a) = \begin{cases} 1 & \text{if } x \in \mathcal{C}_i \wedge x \neq a \\ -\frac{1}{|\mathcal{C}_i|-1} & \text{if } x = a \\ 0 & \text{if } x \notin \mathcal{C}_i \end{cases} \quad (24)$$

where α in the above equation has the same meaning of the one in equation (17).
We assume the sub-sets $\mathcal{C}_{1,2,\dots}$ as given: they can be easily determined by clustering actions with a strong ontological similarity (an extensive review of this topic can be found in [16]). The importance of the information provided by the a priori knowledge w.r.t the one contained in the predictive suffix tree discussed in the previous Section, is determined by tuning weights $w_{0,1,2,\dots}$ (equation (1)), which is the aim of training, see Section II-C.

B. Predicting the waiting times

Since the process governing the evolution of X is stochastic, the time to see again a certain action $a_i \in \mathcal{A}$ can be modelled as a probability distribution function. It is possible to perform a Monte Carlo simulation for collecting a certain number of samples of the latter distribution. Then, the time at which a_i will be done again can be described by the empirical distribution taking into account that samples. For every trial, the series X_n^σ is extended by adding some samples $x_{s1} \triangleright \cdots \triangleright x_{sp}$, such that $x_{sp} = a_i$ and $x_{s1,\cdots,sp-1} \neq a_i$, i.e. the sampling is arrested when finding for the first time the action for which we want to predict the waiting time. Every x_{si} is obtained by sampling from the distribution in equation (1). Every single waiting time T^i_{wait} is obtained by computing the arrival time in x_{sp}. This is done by summing the durations of the intermediate actions $T_{s1,\cdots,sp-1}$ of the intermediate actions:

$$T^i_{wait} = \sum_{i=1}^{p-1} T_{si} \quad (25)$$

Every single T_{si} is generated by sampling from a set of past measured durations, refer to the pipeline in Fig. 2. After performing N trials, the empirical distribution $\{T^1_{wait}, \cdots, T^N_{wait}\}$ is obtained.

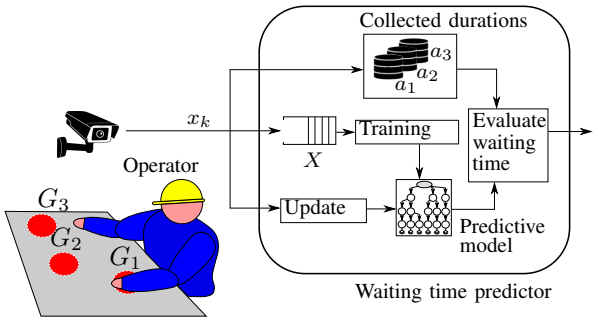

Fig. 2: The pipeline involved in the prediction of waiting times for the actions in $\mathcal{A}$. The predictive model is the one adopted for evaluating the probability expressed by (1), while the collected samples of activity duration are exploited for computing the waiting time T_{wait}.

C. Tuning the model

The weights $w_{0,1,2,\dots}$, see Section II-A, can be determined through learning. In fact, they can be determined in order to maximize the likelihood of X, up to a step K, i.e. considering all the known realizations $x_{1,\cdots,K}$. The logarithm of the joint probability of all the values in X till K (equation (1)) can be determined as the following product:

$$\begin{aligned} L &= log\Big(\mathbb{P}(x_{\sigma+1}|X^\sigma_{\sigma+1}) \cdot \cdots \cdot \mathbb{P}(x_K|X^\sigma_K)\Big) \\ &= \sum_{j=\sigma+1}^{K} \Big(w_0 \Phi_v(X^\sigma_j, x_j|ST) + \cdots \\ \cdots &+ \sum_{Ci} w_{Ci} \Phi_{Ci}(X^\sigma_j, x_j|ST) - log\big(Z(ST)\big) \Big) \end{aligned} \quad (26)$$

Since it is impossible to find the value maximising L in a closed form, a gradient ascend strategy can be adopted. To this purpose, the derivatives $\big[\frac{\partial L}{\partial w_0} \quad \frac{\partial L}{\partial w_{C1}} \quad \frac{\partial L}{\partial w_{C2}} \quad \cdots\big]$ must be evaluated. It is not difficult to prove that their expressions are as follows:

$$\begin{aligned} \frac{\partial L}{\partial w_0} &= \sum_{j=\sigma+1}^{K} \Big(\Phi_v(X^\sigma_j, x_j|ST) + \cdots \\ \cdots &- \sum_{a \in \mathcal{A}} \big(\mathbb{P}(a|X^\sigma_j, ST) \cdot \Phi_v(X^\sigma_j, a|ST) \big) \Big) \end{aligned} \quad (27)$$

$$\begin{aligned} \frac{\partial L}{\partial w_{Ci}} &= \sum_{j=\sigma+1}^{K} \Big(\Phi_{Ci}(X^\sigma_j, x_j) + \cdots \\ \cdots &- \sum_{a \in \mathcal{A}} \big(\mathbb{P}(a|X^\sigma_j, ST) \cdot \Phi_{Ci}(X^\sigma_j, a) \big) \Big) \end{aligned} \quad (28)$$

In principle, it is possible to re-train a model every time a new action x_n is observed for the series X. In such a case, the update and learning (Fig. 2) are done for every step. This is reasonable for the most of real contexts, since the human activity durations (which are in the order of seconds) are higher than the time required for performing the gradient ascent described (in the order of the milliseconds). However,

an approach where learning is done only sporadically is also possible.

III. Experiments

A. Off-line comparisons

With the aim of comparing the developed approach with [6], some off-line simulations were performed, considering the assembly of the emergency button reported in Fig. 3. Steps involved for the completion of a finite product are reported in the same Figure and are made of a series of forks and joints. All the operations in the same fork must be done before the succeeding ones, without a particular order (for instance the screws can be taken before the screwdriver and vice-versa). We created a population of artificial series X, by alternating 20 assembly cycles. Each cycle is a random sequence of operations consistent with the precedence constraints expressed in Fig. 3. Then, an error simulating the non perfect segmentation of human actions was introduced: the 5% of the elements of a X were replaced with random numbers. A total amount of 100 artificial series were generated for producing the results reported in the following. Comparisons are made computing the mean prediction error ε [2] defined as:

$$\varepsilon_k = \frac{1}{|\mathcal{A}|}\left\| \left[\mathbb{P}(x_k = a_1) \cdots \mathbb{P}(x_k = a_m)\right] - \overline{X}_k \right\|_2 \qquad (29)$$

where $\overline{X}_k$ is the distribution modelling the real value seen at step k. For example if $x_k = a_1$, $\overline{X}_k = \begin{bmatrix}1 & 0 & \cdots & 0\end{bmatrix}$. The affine sets are made by considering the actions in the same fork [3] leading to the definition of the following sets: $\mathcal{C}_1 = \{a_1, a_2\}$; $\mathcal{C}_2 = \{a_4, a_5, a_6\}$; $\mathcal{C}_3 = \{a_7, a_8\}$ and $\mathcal{C}_4 = \{a_9, a_{10}\}$.

Fig. 4b reports the results when considering the complete assembly process, while Fig. 4a reports similar statistics but considering the simplified assembly, for which the existence of the affine sets was ignored. As it can be seen, the curve of the mean prediction error of the proposed approach is completely below the one of [6] in Fig. 4b. Moreover, the dispersion is lower (curves of the quantile are closer). Performance are significantly improved when introducing the affine sets and the best performance are achieved when all of them are taken into account. It is important to remark that the approach in [6] would not be able to manage an a-priori knowledge without severely modifying the approach itself. It is interesting to notice that the performance gap is reduced when considering the assembly simplification, Fig. 4a.

As a general consideration, the proposed approach seems to perform well also for low values of σ. Indeed, even with a low order, the suffix tree is able to represent the temporal correlations among the actions, by simply including more branches. For this reason a repeating pattern of actions with a length higher than the model order, would be anyway handled.

B. Experimental validation

Real experiments were conducted by reproducing a realistic human-robot co-assembly, involving the production of torches and clocks. The steps involved are indicated in Fig. 6 [4]. The predictive algorithm discussed so far was adopted for predicting the starting time of the actions assigned to the human. This information was exploited for optimally scheduling the robotic actions, assuming the approach of [5]. In particular, the aim is to minimize the human inactivity times, instructing the robot to execute the actions preceding those assigned to the human and predicted to start in the near future. The starting and the ending of the human actions is recognized by checking when the operator's hands enter into specific areas of the workspace. We omit further details regarding the scheduler and the implementation of the real experiments since are extensively described in [5]. The dual arm YuMi of ABB was employed as robotic mate, while a MICROSOFT Kinect monitored the human in order to detect the starting and the ending of human activities. To this aim, the positions of the operator's hands are tracked during time using the RGB-d image provided by the Kinect sensor. When the hands of the operator enter or go out from a specific spherical area of the workspace [5], an action is recognized to start or be completed. The adopted set-up is reported in Fig. 5. 10 volunteers were enrolled for the experiments, divided into two equally sized groups. The first group (GA) was asked to perform the assigned operations (in the order they prefer) while the scheduler [5], exploiting however the predictions provided by the algorithm described in this paper, controlled the robotic arms. For the other group (GB) the robotic actions were done without exploiting predictions of the human behaviour, since robots were controlled in a reactive way: when the human expressed the intention to begin a certain action [6], its starting was enabled by the robots after performing a specific series of actions.

The predicted waiting time before a new execution of a_1, W_{a1}, for one of the experiments in GA as well as the time at which the operator actually started that action are reported in Fig. 7. The ideal W_{a1} profile is a sawtooth one, assuming null values immediately before the time instants at which a new a_1 is started. As can be appreciated from Fig. 7, the predictions are reliable since the predicted waiting time has a profile very close to the ideal sawtooth one described so far. For this reason, the scheduling approach adopted for GA was able to efficiently produce optimal plans for the robots, reducing the human inactivity times. Comparing the

[2] The prediction errors obtained in each step of the artificial series X are computed using equation (29), leading to a population of errors, whose percentiles are reported in Fig. 4a, 4b and 4c.

[3] This can be also the systematic criterion to follow in industrial contexts.

[4] A video reporting all the activities is available at `https://www.youtube.com/watch?v=ZWKrzrdSlI8`

[5] Such spheres are centred considering the positions of the buffers storing the parts and the working progress involved in the assemblies.

[6] Since one of his or her arm entered in a specific area of the workspace

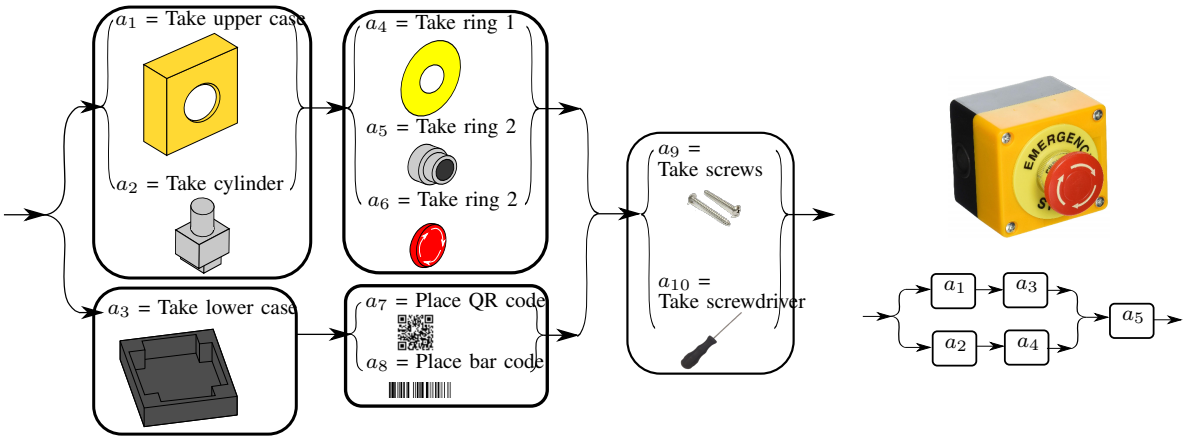

Fig. 3: On the left the sequence of actions required for assembling an emergency button: the actions contained in a box can be done with no particular order, but before the actions contained in the boxes following in the sequence. A total number of 10 actions are needed to finalize the product. The top right part of the Figure reports the emergency button to assemble, while the lower part a simplification of the same assembly, for which the size of set $\mathcal{A}$ is equal to 5.

results coming from GA and GB, we found a clear statistical evidence that the mean cycle time of the experiments in GA is lower than the ones in GB: single-tailed Wilcoxon rank sum returns $r = 0.0251$ when adopting an $\alpha = 5\%$.

IV. Conclusion

This work addressed the prediction of human activities in industrial contexts. A novel algorithm is proposed, factorizing the distribution probability predicting the next action in a time series as a product of terms. The first one is related to a suffix tree, embedding the knowledge of the process acquired during time, i.e. the sequence of past actions seen, while all the others refer to some a-priori information that can be used for improving the prediction process. The approach was proved to perform better than a state of the art algorithm based on a Markovian model through extensive off-line simulations. Moreover, the prediction algorithm proposed was efficiently exploited for enhancing a human-robot co-assembly, allowing the robots to properly adapt and synchronize with the human mate.

Future works should address the development of improved techniques for defining the a priori knowledge leading to the definition of sets $\mathcal{C}$ (see Section II-A.2) which significantly influence the prediction capabilities of the proposed approach, Fig. 4c. Regarding this point, approaches involving the semantic similarities of the actions could be efficiently exploited.

References

[1] Z. Liu and J. Hao, "Intention recognition in physical human-robot interaction based on radial basis function neural network," *Journal of Robotics*, vol. 2019, 2019.

[2] Y. Li and S. S. Ge, "Human–robot collaboration based on motion intention estimation," *IEEE/ASME Transactions on Mechatronics*, vol. 19, no. 3, pp. 1007–1014, 2013.

[3] N. Nikolakis, K. Sipsas, and S. Makris, "A cyber-physical context-aware system for coordinating human-robot collaboration," *Procedia CIRP*, vol. 72, pp. 27–32, 2018.

[4] A. Casalino, A. M. Zanchettin, L. Piroddi, and P. Rocco, "Optimal scheduling of human-robot collaborative assembly operations with time petri nets," *IEEE Transactions on Automation Science and Engineering*, 2019.

[5] A. Casalino, E. Mazzocca, M. G. Di Giorgio, A. M. Zanchettin, and P. Rocco, "Task scheduling for human-robot collaboration with uncertain duration of tasks: a fuzzy approach," in *2019 International Conference on Control, Mechatronics and Automation (ICCMA)*. IEEE, 2019.

[6] A. M. Zanchettin, A. Casalino, L. Piroddi, and P. Rocco, "Prediction of human activity patterns for human-robot collaborative assembly tasks," *IEEE Transactions on Industrial Informatics*, vol. 15, no. 7, pp. 3934–3942, 2018.

[7] H. Koppula and A. Saxena, "Learning spatio-temporal structure from rgb-d videos for human activity detection and anticipation," in *International conference on machine learning*, 2013, pp. 792–800.

[8] B. Reily, F. Han, L. E. Parker, and H. Zhang, "Skeleton-based bio-inspired human activity prediction for real-time human–robot interaction," *Autonomous Robots*, vol. 42, no. 6, pp. 1281–1298, 2018.

[9] K. P. Hawkins, N. Vo, S. Bansal, and A. F. Bobick, "Probabilistic human action prediction and wait-sensitive planning for responsive human-robot collaboration," in *2013 13th IEEE-RAS International Conference on Humanoid Robots (Humanoids)*. IEEE, 2013, pp. 499–506.

[10] Z. Barutcuoglu, R. E. Schapire, and O. G. Troyanskaya, "Hierarchical multi-label prediction of gene function," *Bioinformatics*, vol. 22, no. 7, pp. 830–836, 2006.

[11] K. Li and Y. Fu, "Prediction of human activity by discovering temporal sequence patterns," *IEEE transactions on pattern analysis and machine intelligence*, vol. 36, no. 8, pp. 1644–1657, 2014.

[12] T. Gueniche, P. Fournier-Viger, and V. S. Tseng, "Compact prediction tree: A lossless model for accurate sequence prediction," in *International Conference on Advanced Data Mining and Applications*. Springer, 2013, pp. 177–188.

[13] T. Gueniche, P. Fournier-Viger, R. Raman, and V. S. Tseng, "Cpt+: Decreasing the time/space complexity of the compact prediction tree," in *Pacific-Asia Conference on Knowledge Discovery and Data Mining*. Springer, 2015, pp. 625–636.

[14] K. Kawakami, "Supervised sequence labelling with recurrent neural networks," *Ph. D. thesis*, 2008.

[15] T. Kimoto, K. Asakawa, M. Yoda, and M. Takeoka, "Stock market prediction system with modular neural networks," in *1990 IJCNN international joint conference on neural networks*. IEEE, 1990, pp. 1–6.

[16] V. Cross, "Semantic similarity: a key to ontology alignment." in *OM@ ISWC*, 2018, pp. 61–65.

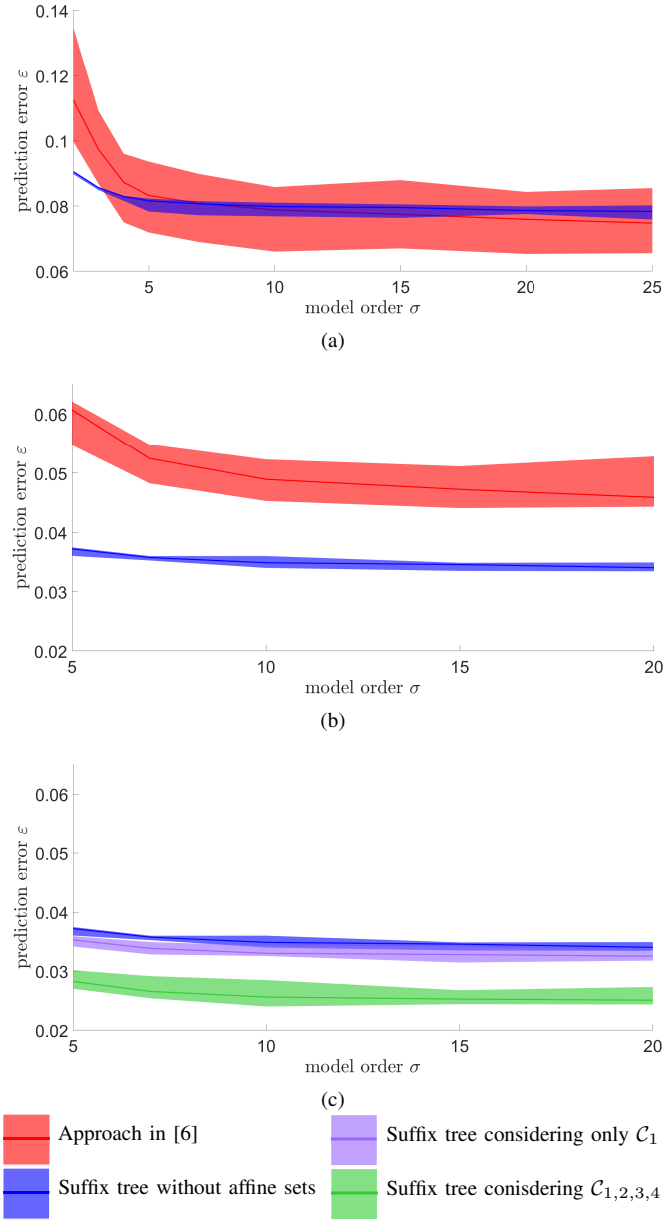

Fig. 4: Statistics of the prediction error obtained from the simulations. The model order σ is the number of preceding actions taken into account for computing the one-step probability prediction, equation (1), while the error ε is the normalized prediction error defined by equation (29). curve of the 50^{th} quartile is inserted into a shaded area delimited by the 80^{th} quartile and the 20^{th} one. Fig. 4b and 4c consider the complete assembly in Fig. 3, while Fig. 4a takes into account the simplification reported in bottom right corner of the same Figure.

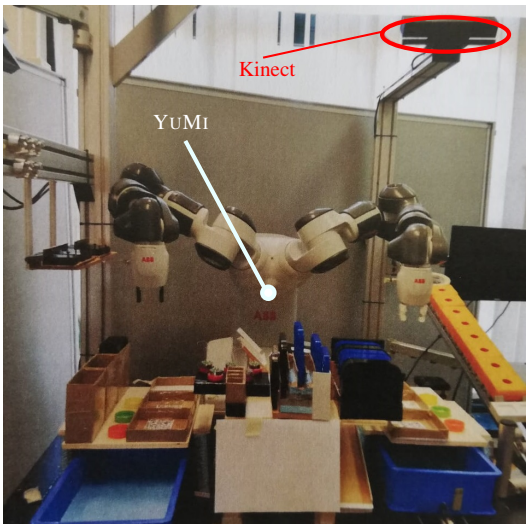

Fig. 5: The robotic cell adopted for the experiments. What is reported, is the point of view of the human inside the cell. The table at the bottom contains several buffers used to store work in progress.

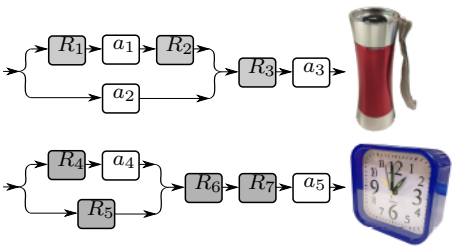

Fig. 6: Actions to perform for finalizing the assemblies of a torch and a clock. Gray shaded boxes refer to actions executed by the robots, all the others, $a_{1,2,3,4,5}$, are the human actions. Notice that $|\mathcal{A}| = 5$.

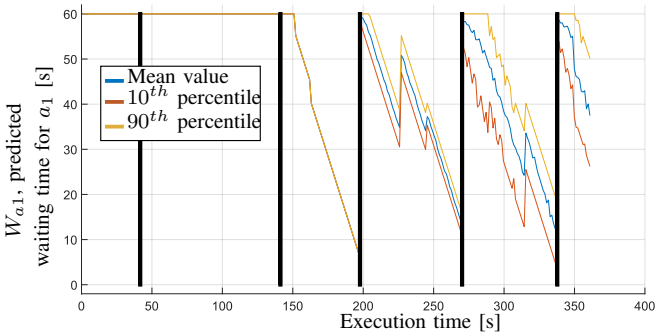

Fig. 7: Evolution of the predicted waiting time W_{a1} for seeing again action a_1. Black vertical lines refer to instants at which the operator started a new execution of a_1. As can be seen, starting from the second execution of a_1, the predicted waiting time begin to be reliable, since decreases quasi-linearly till the time at which a new a_1 is actually started.