# Prediction of human activity patterns for human-robot collaborative assembly tasks

Andrea Maria Zanchettin, *Member, IEEE,* Andrea Casalino, *Student Member, IEEE,* Luigi Piroddi, *Member, IEEE,* and Paolo Rocco *Senior Member, IEEE*

***Abstract*—It is widely agreed that future manufacturing environments will be populated by humans and robots sharing the same workspace. However, the real collaboration can be sporadic, especially in case of assembly tasks which might involve autonomous operations to be executed by either the robot or the human worker. In this scenario, it might be beneficial to predict the actions of the human in order to control the robot both safely and efficiently. In this paper, we propose a method to predict human activity patterns in order to early infer when a specific collaborative operation will be requested by the human and to allow the robot to perform alternative autonomous tasks in the meanwhile. The prediction algorithm is based on higher-order Markov Chains and is experimentally verified in a realistic scenario involving a dual-arm robot employed in a small part collaborative assembly task.**

***Index Terms*—Cognitive Human-Robot Interaction; Intelligent and Flexible Manufacturing; Planning, Scheduling and Coordination;**

## I. Introduction

In the last years, collaborative robots have become faster, smarter, more accurate and reliable. However, challenges remain in adaptability [1], decision making and robustness to changing and uncertain situations, especially when a continuing interaction with the human co-worker is expected. Apart from safety-related research studies [2], [3], which aim at minimising the interference between the human and the robot, cognitive algorithms proved to be capable of enhancing the effectiveness of the collaboration [4]. Such algorithms allow robots to understand the behaviour of their fellow human team-mates in order to anticipate, and adapt to them, [5]. A significant literature is focused on the task assignment problem. For example, Chen et al. [6] describe a genetic algorithm for a collaborative assembly station which minimises the assembly time and costs. In [7], a trust-based dynamic subtask allocation strategy for manufacturing assembly processes has been presented. The method, which relies on a Model Predictive Control (MPC) scheme, accounts for human and robot performance levels, as well as on their bilateral trust dynamics. Furthermore, in [8], the authors proposed a multi-layered planner for task allocation, sequencing and execution using AND/OR graph and $A^*$ graph search. Similarly, in [9] Tsarouchi et al. proposed an intelligent decision-making method that allows human-robot task allocation according to their capabilities.

By taking inspiration from real-time processor scheduling policies, Gombolay et al. [10] developed a multi-agent task sequencer, where task specifications and constraints are solved using a MILP (Mixed Integer Linear Programming) algorithm, showing near-optimal task assignments and schedules. A similar approach has been also derived in [11]. Recently, in [12] an approach based on game theory has been proposed to estimate the objective of the human, through the measured interaction force, and to adapt the robot objective accordingly.

In the context of the so called *Industry 4.0*, it is paramount to develop digital models of each agent taking part in the manufacturing process, [13], should this agent be a robot, a machinery, or the human operator. Tracking humans activities and predicting the best instant when the robot should be available to give assistance is a crucial prerequisite for a fluent collaboration. Hawkins et al. [14] developed an inference mechanism based on Hidden Markov Models (HMMs) allowing the robot to predict when particular robot actions would be appropriate, based on the current state of the human worker. HMMs have been also adopted in [15] to recognise and label sequences of activities based on occupancy grids. The approach capitalises on the multi-modal perception algorithm discussed in [16]. With the aim of reducing the worker's waiting time, Kinugawa et al. [17] developed an online learning algorithm to feed an adaptive task scheduling system for the collaborative robot. Other approaches based on neural networks [18], or Dynamic Bayesian Networks (DBN) [19]–[21] have been developed to investigate the mutual adaptation of hybrid human-robot teams by modelling motion patterns. In [22], Li et al. proposed a framework based on variable order Markov models (VOMM) to predict activity patterns using causal relationships between actions. Variable order stochastic automata were also used before in [23]. Other works focusing on high-order stochastic processes, but not applied to robotics, can be found in [24]–[26]. This work presents a method to model and predict human activity patterns, in order to endow the robot with all the information needed to take the best action. In a nutshell, the method developed in this research allows to give answers to the following questions:

1) Which activity is the human more likely to perform next?
2) What is the time when an activity requiring assistance is expected to be initiated by the human?

In particular, we focus on an assembly task where both the human and the robot have individual subtask assignments and

The authors are with Politecnico di Milano, Dipartimento di Elettronica, Informazione e Bioingegneria, Piazza Leonardo Da Vinci 32, Milano, Italy e-mail: (firstname.lastname@polimi.it). The original method described in this work is protected by a patent application, IT 102018000006156, filed by Politecnico di Milano and Smart Robots s.r.l..

a joint action has to be performed in order to finalise the assembly. The collaborative operation requires the two agents to be available at the same time in order to be initiated, while the two individual operations can be initiated at any time instant. Based on the need for the robot to predict the human activity pattern, this paper introduces a method to infer the waiting time for a certain action to be performed by the human. In the reported example, this action would be the collaborative operation. Figure 1 reports a pictorial representation of the dependencies between the agents (human and robot) and the developed algorithms. In particular, a prediction mechanism is used both to infer the human's current activity and to predict his/her behaviour in terms of the next assembly steps to be performed. The outcome of this algorithm is then used to feed a scheduling algorithm, that allows the robot to decide which action to perform in order to be at the same time assistive and productive. The basic assumption underlying this work is that the behaviour of the robot has little influence on the (future) decisions of the human. Moreover, as we believe that workers still represent an added value at the shop-floor, the method we propose allows the robot to adapt to the human behaviour and pace, and not vice-versa. The developed method is verified experimentally within a collaborative assembly task.

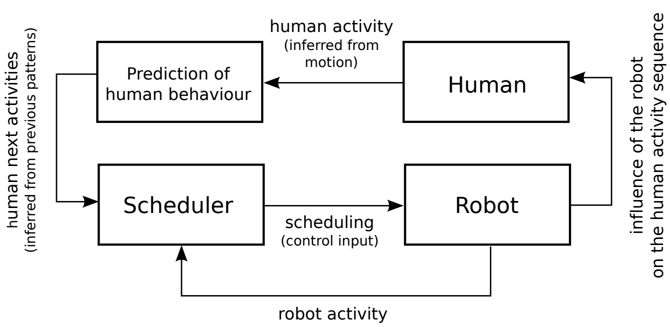



Fig. 1. A pictorial representation of the interdependencies within a collaborative workstation.

The remainder of this work is organised as follows. Section II describes the working assumptions and the developed methodology. Section III reports a comparison with state of the art methods. Section IV introduces the use case, the validation setup, the implementation details and discusses the outcome of the experiments. Section V offers some concluding remarks.

## II. Modelling and Prediction of Activity Patterns

In typical collaborative assembly stations, positions are related to stocks where parts or tools are located. Figure 2 sketches a generic collaborative station. The robot and the human worker are sharing the same workspace and collaborate in the advancement of the task. In this situation, it is crucial for the robot to synchronise with its fellow human worker in order to minimise the cycle time of the operation. The algorithm developed in this paper addresses this issue and is composed of two main parts. The former takes the sequence of all labeled activities up to the current time instant and models them in terms of a stochastic process. The latter tries to infer its future evolution in time. In the following, the approach adopted in this paper is further detailed.

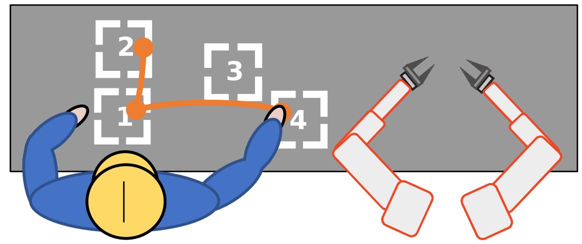



Fig. 2. Example of collaborative assembly layout. The human operates in four possible positions, numbered from 1 to 4. The robot task must be synchronous with the one assigned to the human. For example, the robot should promptly pick up a part dropped by the human in position 4.

### A. *Modelling activity patterns*

Human assembly sequences usually form quasi-repetitive patterns. In other words, the sequence of human activities can be modelled through a time series, which is the output of a certain dynamic process. Assuming the set of available activities to be enumerable and finite, i.e $\forall k, A_k \in \mathbb{A} = \{1, 2, \ldots, m\} \subset \mathbb{N}$ where $A_k$ is the ongoing activity at discrete time instant $k$, the behaviour of the human fellow co-worker can be modelled through the following discrete-time process

$$\begin{aligned} A_{k+1} &= f\left(A_k, A_{k-1}, A_{k-2}, \ldots, A_{k-n}\right) \\ t_{k+1} &= t_k + g\left(A_k\right) \end{aligned} \tag{1}$$

where $t_k \in \mathbb{R}^+ \cup \{0\}$ represents the time instant corresponding to the transition from $A_{k-1}$ to $A_k$ and $g\left(a\right) = T^a > 0$ is the duration of activity $a \in \mathbb{A}$. In the prediction model (1) we do not explicitly account for robot actions and their influence on the sequence of human actions. However, as the parameters of model will be constantly updated using online data, the identified model will be be indirectly influenced by robot actions in case they have influence on human ones. A possible time evolution of the process is represented in Fig. 3. The dynamic system in (1) is more easily identifiable in two stages. First, we address the identification of the underlying discrete event process governing the evolution of activities (i.e. the first equation), regardless of their duration. Then, we provide a model for the duration of the activities, that will be described later on.

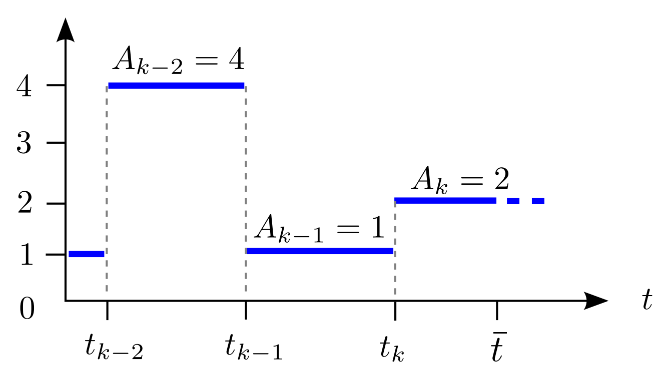



Fig. 3. Example of behaviour of the human: $t_k$ represents the time instant corresponding to the activation of activity $A_k$, while $\bar{t}$ represents the current time-stamp.

Given the stochasticity of the underlying discrete process governing the sequence of activities, we describe the probability distribution of a certain activity at discrete-time $k+1$, given the previous history. Common manufacturing or assembly activities are difficult to model as Markov Chains, i.e. implying that the next activity only depends on the current one. Better results can be achieved if the human behaviour is modelled as a higher-order Markov Chain. We are interested in computing the probability associated to the next activity (or more in general to the next sequence of activities, if evaluated recursively) given the sequence of the previous ones, i.e.

$$\mathcal{P}\left(A_{k+1}=a \,|A_k=k_0, A_{k-1}=k_1, \ldots, A_{k-n}=k_n\right). \quad (2)$$

Differently from usual Markov Chains, generic higher-order Markov Chains require $m^{n+1}(m-1)$ parameters to be estimated, resulting in an exponential complexity with respect to the order of the stochastic process to be identified.

The work from Raftery [24], [25] proposed an efficient way to describe higher-order Markov Chains using Mixture Transition Distribution (MTD) models. Specifically, the probability distribution in (2) is represented as

$$\begin{aligned}&\mathcal{P}\left(A_{k+1}=a \,|A_k=k_0, \ldots, A_{k-n}=k_n\right) \approx \\ &\approx \sum_{i=0}^{n} \lambda_i \mathcal{P}\left(A_{k+1}=a \,|A_{k-i}=k_i\right)\end{aligned} \quad (3)$$

hence as a convex combination (mixture) of multiple-steps transition probabilities. This model, that corresponds to usual Markov Chains for $n=0$, requires only $m^2(n+1)$ parameters. Using the canonical representation for categorical sequences, the sequence $\{A_k\}$ can be written in terms of state vectors $\boldsymbol{X}_k$, which are columns of zeros, except for a 1 in row $A_k+1$. For example, if $A_k=2$, $\boldsymbol{X}_k$ would correspond to $\boldsymbol{X}_k=\begin{bmatrix}0 & 1 & 0 & \ldots\end{bmatrix}^T$.

According to the work from Raftery, [24], a prediction of the probability distribution $\hat{\boldsymbol{X}}_{k+1}$ at time $k+1$ can be computed as

$$\hat{\boldsymbol{X}}_{k+1}=\sum_{i=0}^{n} \lambda_i \boldsymbol{Q}_i \boldsymbol{X}_{k-i} \quad (4)$$

where $m \times m$ matrix $\boldsymbol{Q}_i$ denotes the $i$-steps transition probability matrix that can be simply evaluated through count statistics. As for the online estimation of the weights $\lambda_i$ from data, differently from Ching et al. [26] who adopted an estimate of the stationary distribution $\boldsymbol{X}_\infty$, we introduce a data-driven procedure. Using all the available evaluations until the present time instant one can evaluate the squared norm of the prediction error, i.e. $\left\|\hat{\boldsymbol{X}}_{k+1}-\boldsymbol{X}_{k+1}\right\|^2=\left\|\sum_{i=0}^{n} \lambda_i \boldsymbol{Q}_i \boldsymbol{X}_{k-i}-\boldsymbol{X}_{k+1}\right\|^2$. By stacking all these evaluations available for different values of $k$, i.e.

$$\begin{bmatrix}\boldsymbol{Q}_1\boldsymbol{X}_{k-2}\lambda_1+\boldsymbol{Q}_2\boldsymbol{X}_{k-3}\lambda_2+\ldots \\ \boldsymbol{Q}_1\boldsymbol{X}_{k-3}\lambda_1+\boldsymbol{Q}_2\boldsymbol{X}_{k-4}\lambda_2+\ldots \\ \vdots\end{bmatrix}-\begin{bmatrix}\boldsymbol{X}_k \\ \boldsymbol{X}_{k-1} \\ \vdots\end{bmatrix}=\boldsymbol{A}\boldsymbol{\lambda}-\boldsymbol{b}$$

the optimal solution for the $\lambda_i$'s parameters can be obtained by a non negative least-squares problem of the following type:

$$\min_{\boldsymbol{\lambda}} \|\boldsymbol{A}\boldsymbol{\lambda}-\boldsymbol{b}\|^2 \text{ subject to } \sum_{i=0}^{n} \lambda_i=1, \text{ and } \lambda_i \geq 0 \quad (5)$$

where the column vector $\boldsymbol{\lambda}$ collects all the unknown parameters $\lambda_i$, while the regression matrix $\boldsymbol{A}$ and vector $\boldsymbol{b}$ can be simply evaluated from data.

### *B. Prediction of human activities and their duration*

So far, the human behaviour has been modelled as a sequence of activities, regardless of their duration. In order to predict in the most effective way when a certain activity is undertaken by the human, it is necessary to account for their time durations as well. We here assume that the duration of activity $a \in \mathbb{A}$, i.e. $T^a$, can be modelled as a stochastic variable with a strictly positive lower bound, i.e. $T^a \geq T^a_{min} > 0$.

In order to estimate the waiting time needed for the certain activity $a$ to show up, say $\tau^a$, we can combine this information with the one described in the previous subsection. In particular, at the present continuous time instant $\bar{t}$, given the sequence of the last activities (possibly including the currently running one) $A_k, A_{k-1}, \ldots, A_{k-n}$, we would like to estimate the probability distribution of the waiting time for the beginning of a certain activity $a$, i.e. $\mathcal{P}\left(\tau^a \leq t \,|A_k, A_{k-1}, \ldots, A_{k-n}\right)$.

The key idea is to construct the reachability tree of the higher-order Markov Chain described earlier and evaluate the time spent to traverse each possible branch which terminates with the desired activity $a \in \mathbb{A}$. To do so, as the reachability tree is, in principle, infinite, we first define a prediction horizon $\Delta T$ meaning that the given probability will be computed up to the instant $t=\bar{t}+\Delta T$.

The probability associated to each branch can be simply computed using (3) by multiplying the probability of each arc of the branch, i.e.

$$p_{branch}=\prod_{(i,j)\in branch} p_{(i,j)}.$$

As for the waiting time associated to each branch $\tau_{branch}$, this is simply the sum of the duration of each activity $T^a$, i.e

$$\tau_{branch}=\sum_{j:(i,j)\in branch} T^j.$$

Notice that the elapsed time of the ongoing activity as well as the tails of the activities exceeding the prediction horizon $\Delta T$ have to be removed. The time associated to each branch is computed as the sum of stochastic variables which are not, in general, identically distributed. Moreover, neither the associated distribution nor its parameters are a priori known. Since it may turn out to be difficult to select a model to describe the probability distribution of the duration of each activity, in this work we used directly the statistics associated to recently acquired samples. Figure 4 also reports an example of distribution of duration of a certain activity.

Finally, given the distributions of the times associated to each branch, the overall distribution of the waiting time of the activity $a$ can be simply computed as a weighted sum of the waiting times associated to each branch, i.e.

$$\begin{aligned}&\mathcal{P}\left(\tau^a \leq t \,|A_k, A_{k-1}, \ldots, A_{k-n}\right)= \\ &=\sum_{branch} p_{branch}\mathcal{P}\left(\tau_{branch} \leq t\right).\end{aligned} \quad (6)$$

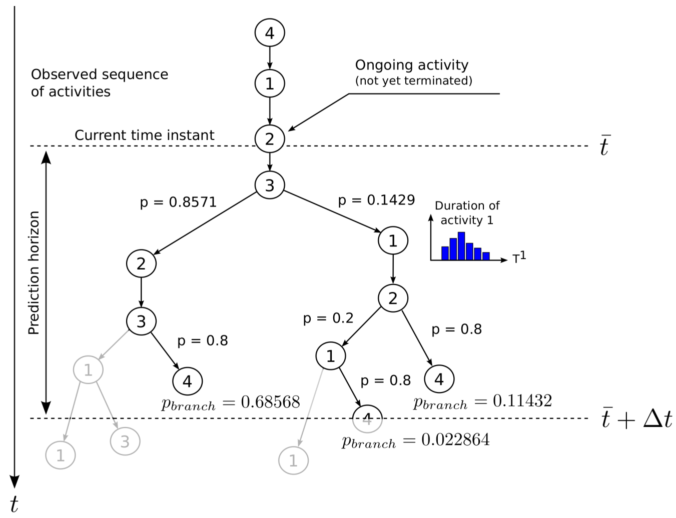



Fig. 4. Example of prediction of human future activities. The transition probabilities associated to each arc are evaluated using (3). The lower bounds on the duration of each activities are used to prune branches of the tree that surely exceed the given prediction horizon $\Delta T$. For all the remaining branches (three in the reported example), the corresponding distributions of waiting times $\tau_{branch}$ are computed and used within (6) to estimate the distribution of the waiting time needed for a certain activity to show up. In this example, the probability distribution of the waiting time of activity 4, i.e. $\tau^4$, is computed.

**Algorithm 1** Reachability Tree Expansion

```
procedure TreeExpand(Q_i, λ_i, ΔT, a ∈ 𝔸)
    while true do
        set the root as an expanded leaf;
        if all leaves expanded then
            return;
        else
            pick a non expanded leaf;
            if current leaf corresponds to activity a then
                mark current node as expanded;
            else
                compute (4);
                append m leaves to the current node;
                for each leaf do
                    set τ_branch = Σ_{j:(i,j)∈branch} T^j_min;
                    evaluate p_branch;
                    if τ^min_branch > ΔT or p_branch < ε then
                        mark current node as expanded;
```

Figure 4 reports an example of the application of the developed method, while Algorithm 1 summarises the steps required to construct the reachability tree.

When the described algorithm is run continuously at a certain frequency, an updated estimation of the waiting time for a certain activity to arrive is available at each iteration. As an example, Fig. 6 reports the typical behaviour of the output of the algorithm corresponding to a certain activity sequence.

The overall block diagram is reported in Fig. 5.

## III. Comparison with existing methods

Markov Chains, or in general HMMs, have been extensively used in the literature to model and predict human behaviour in collaborative tasks, [14], [15]. A possible drawback, which is inherent in the Markov's assumption, is that they restrict the modelling capabilities to strictly Markovian processes, without the ability to capture periodic or repetitive patterns of actions which are common in assembly stations. In fact, in manufacturing environments (and especially in assembly) the next activity to be performed does not depend solely on the current one (Markov's assumption). In other words the process has a memory longer than one step (the whole sequence of assembly steps). Also, the authors of [22] reported the same limitation of HMMs to model long-term causality dependencies between actions. In order to evaluate the performance in terms of prediction error of the proposed algorithm, the sequence $1 \to 2 \to 3 \to 1 \to 2 \to 4 \to 3 \to 5$ has been repeated for 75 times, and random mutations with probability $3\%$ have been applied. The same resulting sequence has been processed with different algorithms. Figure 7 reports the average 1-step ahead prediction error for the analysed methods. It can be noticed that the proposed method sensibly outperforms the others, especially for high lengths of the FIFO buffer. The reason is due to the least-squared optimisation method which provide robustness to the algorithm in case of quasi-periodic patterns. Moreover, the limited prediction capabilities of Markov Chains in case of higher-order causality can be also appreciated. Finally, it is worth noticing that the memory storage required by the VOMM method proposed in [23] is linear with respect to the length of the FIFO buffer, and significantly higher then the one required by the methods based on Markov chains, which, in turn, does not depend on the length of the FIFO buffer. It is worth noticing that, in order to obtain a good performance, the length of the FIFO buffer should be at least twice as long as the order of the system, see (1). On the other hand, for better performance in case of a sudden change of pattern, the length of the FIFO buffer should be kept at a minimum. It follows that a rough knowledge of length of the typical pattern is necessary for the algorithm to achieve the best prediction accuracy.

A temporal Bayesian Network is used in [19] to infer the best time for the robot to act, so that the waiting time is minimised for both the human and the robot. While similar in its objectives, [19] assumes that the conditional dependencies between the human actions are known, while in our work they are learnt and constantly updated using previous observations. In addition, the work in [19] assumes Gaussian distributions for the durations of the activities. This appears to be a limiting assumption, often violated in practice. In our approach we adopt instead a data-driven approach by storing all the durations in a database as statistical populations, see also Fig. 5.

Finally, the work in [17] introduced a model for the prediction of the worker's arrival time at a certain working position within an automotive assembly process. Based on this prediction, a collaborative robot decides the appropriate timing of its assistive action. Though similar in purpose, our work is more focused on the small parts assembly problem, which does not require significant movements of the operator within the assembly station. Moreover, while the work by Kinugawa et al. describes a method to predict the end time of the

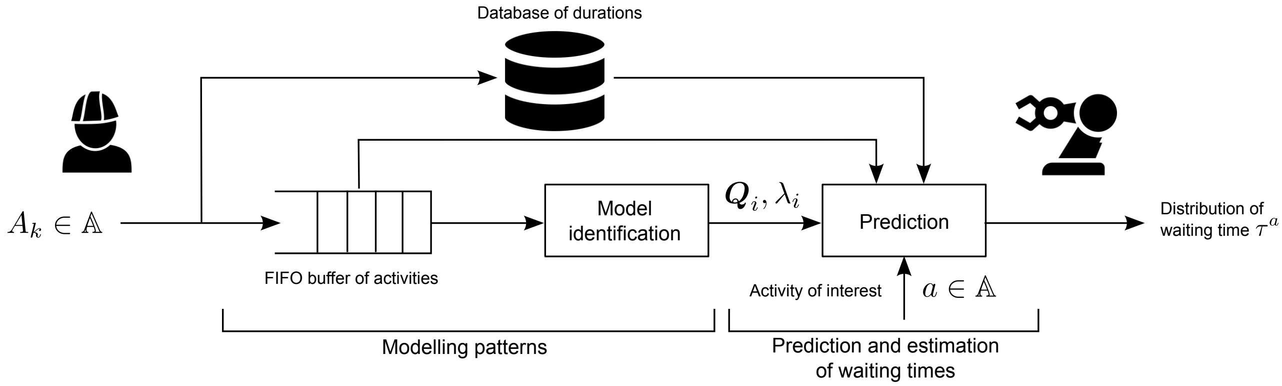



Fig. 5. Block diagram of the overall algorithm showing its two major components: (1) modelling patterns, and (2) prediction and estimation of waiting times.

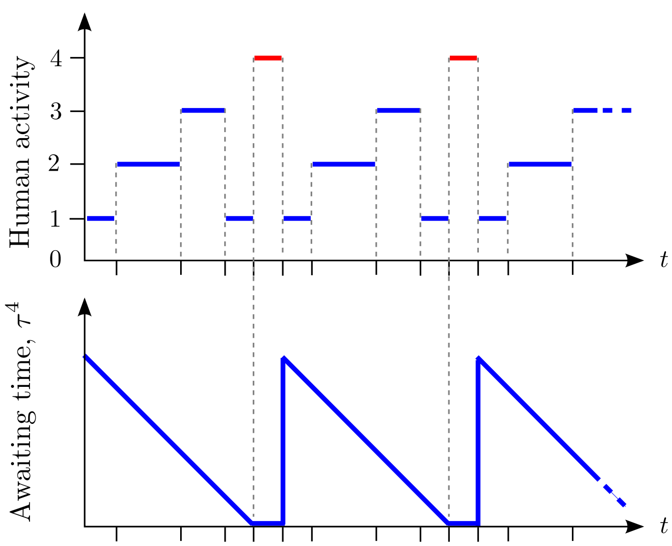



Fig. 6. Example of sequence of activities (top) and corresponding typical behaviour of the estimate of the waiting time of activity 4, i.e. $\tau^4$, (bottom) evaluated and continuously updated during time.

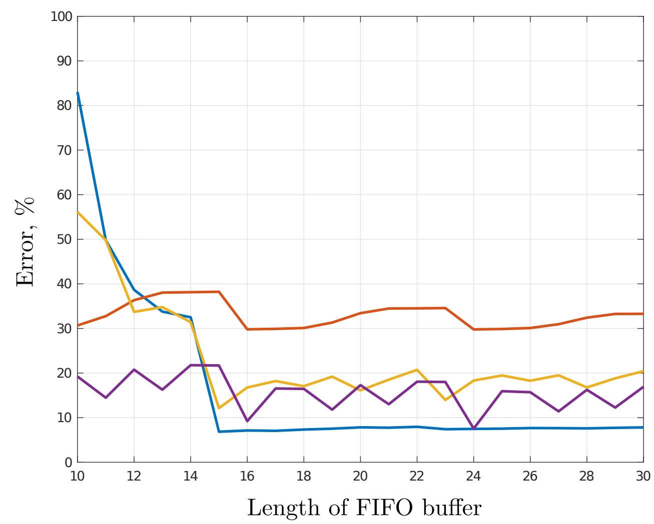



Fig. 7. Comparison of different methods in terms of prediction error: the presented algorithm ($n = 7$, blue), the VOMM method proposed in [23] (purple), a Markov Chain model ($n = 0$, red), a higher-order Markov Chain models trained with the algorithm proposed by Ching et al. in [26] ($n = 7$, yellow).

ongoing operation, the present work is focused on a long-term prediction of human behaviour and not just to the ongoing activity.

## IV. Use-case and experiments

In order to test the effectiveness of the proposed approach, a realistic use case of human-robot collaborative assembly has been set up. The collaborative workspace consists of an ABB dual-arm robot YuMi equipped with a suction tool and a parallel gripper. A Microsoft Kinect depth camera is used to acquire the positions of the human's hands in order to evaluate the sequence of operations. The human and the robot actively cooperate to perform the assembly of a PCB board to be accommodated within an IP 54 plastic enclosure. A picture of the experimental setup is shown in Fig. 8.

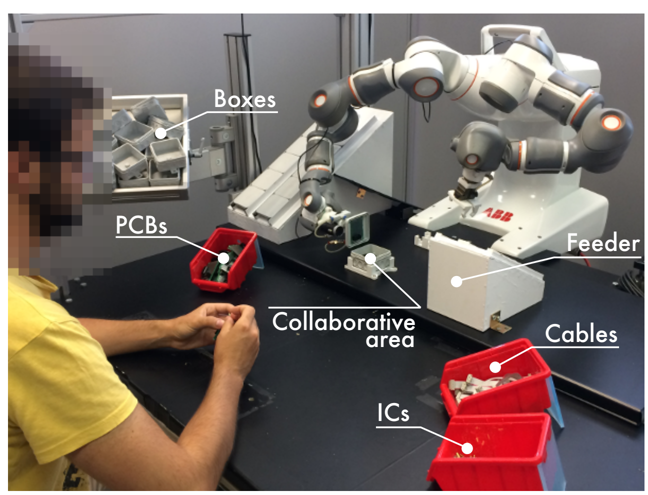
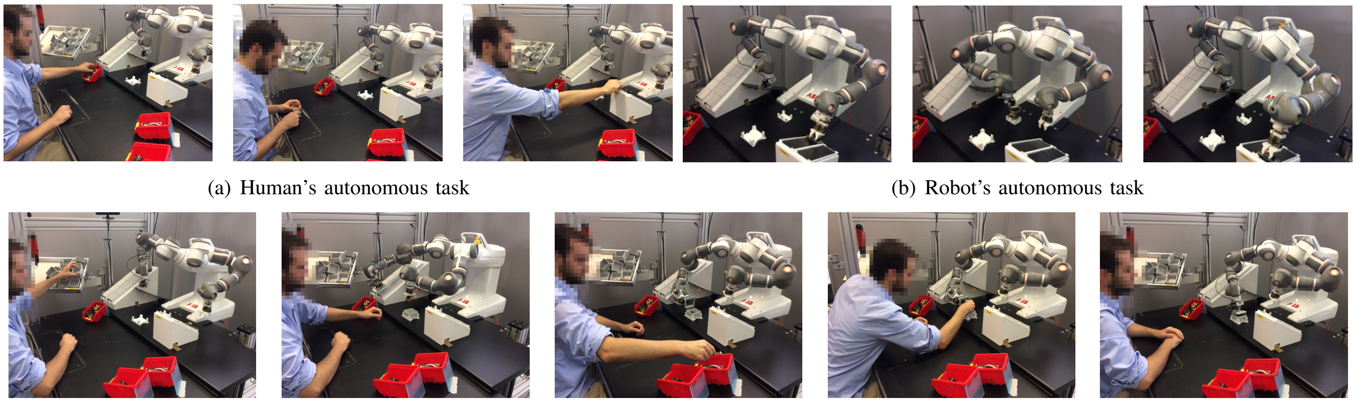



Fig. 8. Layout of the experimental setup: the human can access six stations, the central one being dedicated to the collaboration with the robot.

### A. Task description and implementation

The human is responsible for an autonomous task which consists in assembling an integrated circuit into a socket already soldered onto a PCB. In turn, the robot is responsible for verifying the quality of the resulting assembly. The different



(a) Human's autonomous task

(b) Robot's autonomous task

(c) Collaborative task

Fig. 9. Different phases of the assembly procedure. **IC insertion** (top left): the human takes a PCB board from the red box on the left and an IC from the red rightmost box, inserts the IC in the pre-soldered socket, and finally fills the feeder. **Quality check** (top right): the robot takes a PCB from the feeder, accommodates it within a fixture, then it takes a picture of the PCB using the in-hand camera, and finally drops it on the feeder. **Flat assembly and finalisation** (bottom): the human takes a plastic enclosure from the left tray and places it in the fixture in front of the robot within the collaborative area, the robot picks a verified PCB and places it inside the enclosure, the human takes a flat cable from the right red box, meanwhile the robot takes the cap from a feeder and assists the human while fixing the cable on it, the robot accommodates the cap on the enclosure and finally stores the finished part.

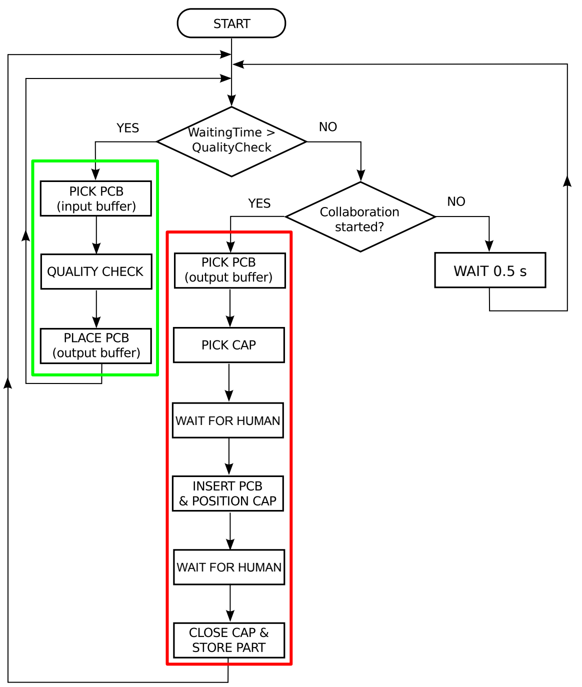



Fig. 10. Workflow of the robot program: based on *WaitingTime*, i.e. the $p$-percentile $t_p$ returned by the algorithm, the first decision the robot takes is whether to wait for the human to initiate the collaborative operation (on the right, grey box) or to start the autonomous subtask (on the left, green box). The collaborative operation (in the middle, orange box) starts when initiated by the human.

phases of the assembly procedure are shown in Fig. 9. The method developed in this work and described in Section II is able to compute the probability distribution of the waiting time for a certain human activity to show up. Should this activity require some kind of assistive behaviour from the robot, it is essential for the robot task planner to know whether a subtask can be initiated or not. Within the present use case, the robot is responsible for an autonomous activity (quality check) but also for being of assistance to the human in holding, like a third hand, the cap while the operator is fixing the flat cable. Therefore, the waiting time for the collaborative operation is constantly estimated and, when needed, compared to the execution time of the quality check. In particular, the algorithm described in Section II computes the distribution of the waiting time and returns a specified $p$-percentile $t_p$.

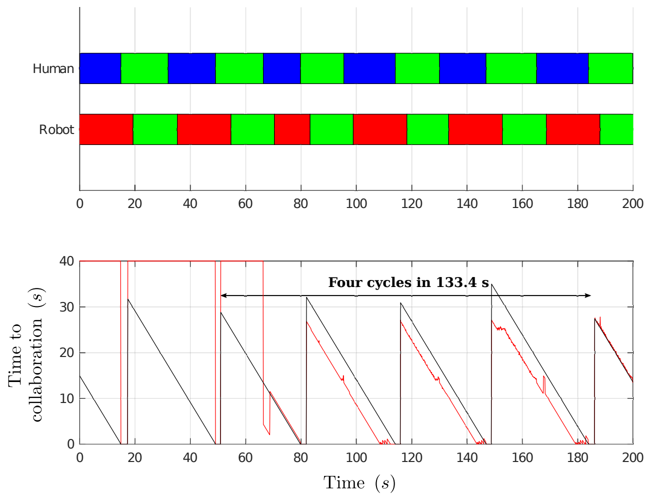
If the time remaining before the collaborative operation is larger than the time the robot needs to complete the quality inspection of one part, the robot initiates its autonomous task. Otherwise, the robot waits in order to be ready to assist the human during the collaborative operation. This behaviour has been coded within the robot programming language and the corresponding flow chart is reported in Fig. 10.

The algorithm to predict the human activity patterns, sketched in Fig. 4, has been coded within an INTEL NUC Core i3 with 16 $GB$ of memory and runs at the same frequency of the KINECT skeleton tracking routine (30 $Hz$). At the same frequency, the waiting time for the collaborative action is evaluated and communicated, through an Ethernet connection to the robot controller which in turn runs the algorithm shown in Fig. 10, coded in its native language. Every time a human activity is terminated, the higher-order Markov Chain model is updated by computing the parameters $\boldsymbol{Q}_i$ and $\lambda_i$ in (4), using the optimisation algorithm described in Section II solved by the open-source library QuadProg++.

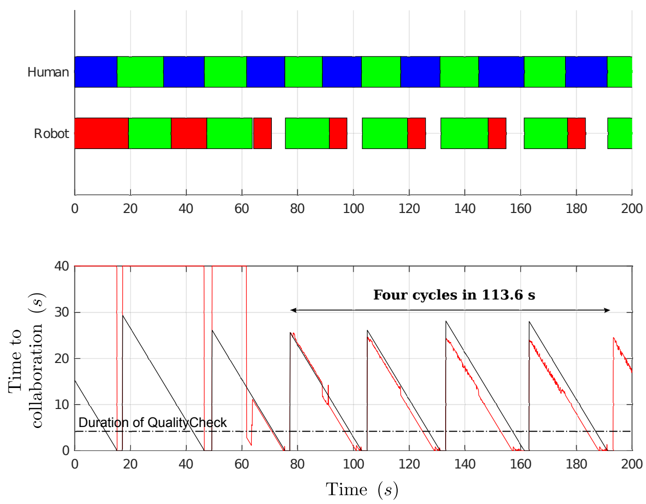
## B. Experiments and discussion

For validation, two different experiments have been run. In the first one, the algorithm developed in this work has been enabled. The second experiment has been run for comparison: the algorithm developed in this paper has been disabled and the robot keeps executing its autonomous task, unless the human operator has already initiated the collaborative operation. In other words, the robot implements a purely reactive strategy.

Figure 11(b) reports the sequence of activities performed by the robot and the human, together with the estimate of the




(a) Reactive approach




(b) Proactive approach

Fig. 11. Execution of the collaborative assembly experiment with the reactive (top) and the proactive (bottom) approach. The top Figures shows the sequence of activities of the human left hand and of the robot (blue and red represent autonomous activities, while the collaborative operation is marked in green). The bottom figures show the predicted time to collaboration (picking a box from the left tray, see Fig. 8) as compared to the ground truth (black).

waiting time until the request for collaboration. As one can see, after a training phase lasting around 60 $s$, which is required for the method to collect enough data to solve the optimisation problem in (5), the robot is able to schedule the right operation, i.e. to wait for the human to initiate the collaborative task instead of initiating its autonomous assignment, which would have caused the human to wait before being assisted.

For comparison, during the second experiment the same assembly task is executed without the proposed algorithm, i.e. using a purely reactive approach. Differently from the previous case, the robot is always assigned to the autonomous task, unless the human has already initiated the collaborative part. As one can notice from Fig. 11(a) the overall execution of the last complete four assembly cycles takes around 20 $s$ more (133.4 $s$ vs. 113.6 $s$ with the proposed approach), which corresponds to an increase of 17% in terms of throughput, thus confirming that the proactive behaviour developed in this work allows for a more fluent and efficient task execution. Moreover, in collaborative applications, and because of safety limitations, robots are typically slower than caged industrial manipulators. Thanks to the developed technology, the possibility to reduce the cycle time and thus improve the efficiency of the assembly cycle would further boost the return on investment (ROI) of collaborative robots. As a further confirmation, Fig. 12 reports

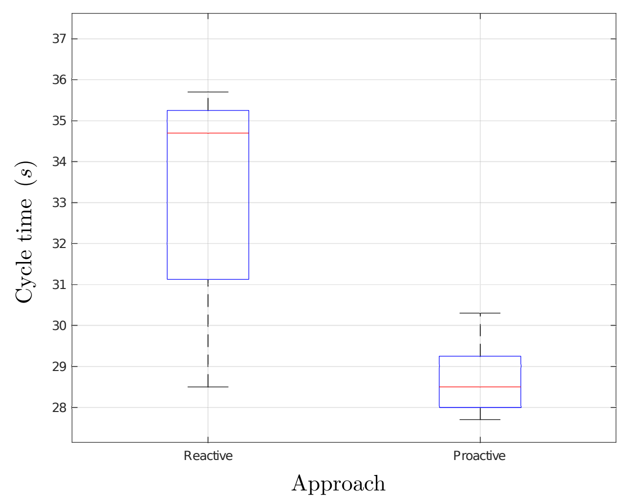



Fig. 12. Distribution of cycle times of the whole assembly sequence with the two approaches. The proactive approach described in this paper is responsible of a higher throughput as well as a reduced variability in cycle times.

the distribution of the cycle time corresponding to the reactive and the proactive approaches. As already stated, the proactive approach proposed in this paper outperforms the reactive one in terms of a reduced cycle time (Wilcoxon signed rank right-tail test, $r = 0.9848$), Fig. 12 also shows that the variability can be reduced (F-test, $r = 0.9705$) by adopting a proactive behaviour. Overall, the prediction algorithm detailed in Section II performs well in predicting the time before the next demand for collaboration from the human. As one can see from Fig. 11(b), the estimated waiting time is slightly underestimated with respect to the ground truth value, and results in a saw-toothed profile with respect to time, as expected.

So far the, a *one piece flow* pattern has been adopted by the human operator. In different production scenarios, some other patterns can be also adopted. Another experiment has been performed to test the capabilities of the algorithm. In particular, the human adopted pattern which consisting in two consecutive IC insertions and two consecutive collaborative operations. The results are reported in Fig. 13. It is worth noticing that the duration of the time interval between the beginning of two consecutive collaborative operations now assumes a bimodal distribution. For this reason, any other approach based solely on this information will be surely less precise than any other method that attempts to model what happens between two consecutive events with a higher granularity. In Fig. 13 a comparison between the method developed in this paper and a purely data-drive approach is reported. The latter is obtained by collecting the time intervals between two consecutive requests for collaboration (up to the present time instant), and the prediction is made extracting the same percentile from the obtained distribution. As one can notice the proposed method significantly outperforms the other in predicting the remaining time before the next demand for collaboration from the human operator.

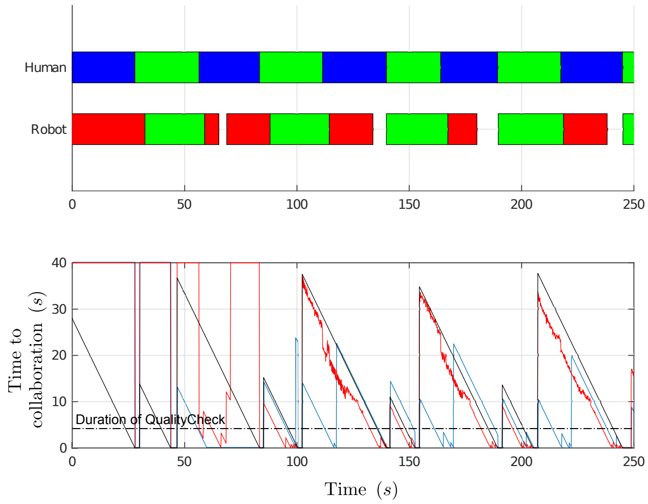



Fig. 13. Execution of the collaborative assembly experiment with the proactive approach. Here the human adopts a different pattern which consists in two consecutive IC insertions and two consecutive collaborative operations. The notation is identical to the one of Fig. 11, except from the blue curve which represents a purely data-drive approach.

## V. Conclusions

An algorithm to recognise and predict human activity patterns during collaborative assembly operations has been proposed. Its output consists in an estimate of the remaining time until a certain operation, which requires assistance from the robot, will be performed again by the human. This estimate is fed to the robot controller, where an algorithm is responsible for the scheduling of robot tasks. In particular, the robot can decide whether to initiate an autonomous task or to wait in order to be ready to promptly assist the human, when needed. For validation, a realistic demonstration consisting in a collaborative assembly task of small parts has been setup. Thanks to the developed algorithm, each cycle of the collaborative task requires a significantly reduced amount of time, as compared to the case when the algorithm is not enabled.

## References


[1] C. Yang, C. Zeng, Y. Cong, N. Wang, and M. Wang, "A learning framework of adaptive manipulative skills from human to robot," *IEEE Transactions on Industrial Informatics*, pp. 1–9, 2018.

[2] F. Flacco, T. Kroger, A. De Luca, and O. Khatib, "A depth space approach to human-robot collision avoidance," in *Robotics and Automation (ICRA), 2012 IEEE International Conference on*. IEEE, 2012, pp. 338–345.

[3] A. M. Zanchettin, N. M. Ceriani, P. Rocco, H. Ding, and B. Matthias, "Safety in human-robot collaborative manufacturing environments: Metrics and control," *IEEE Transactions on Automation Science and Engineering*, vol. 13, no. 2, pp. 882–893, 2016.

[4] G. Hoffman and C. Breazeal, "Cost-based anticipatory action selection for human–robot fluency," *IEEE Transactions on Robotics*, vol. 23, no. 5, pp. 952–961, 2007.

[5] T. Iqbal, S. Rack, and L. D. Riek, "Movement coordination in human–robot teams: A dynamical systems approach," *IEEE Transactions on Robotics*, vol. 32, no. 4, pp. 909–919, 2016.

[6] F. Chen, K. Sekiyama, F. Cannella, and T. Fukuda, "Optimal subtask allocation for human and robot collaboration within hybrid assembly system," *IEEE Transactions on Automation Science and Engineering*, vol. 11, no. 4, pp. 1065–1075, 2014.

[7] S. M. Rahman, B. Sadrfaridpour, and Y. Wang, "Trust-based optimal subtask allocation and model predictive control for human-robot collaborative assembly in manufacturing," in *ASME 2015 Dynamic Systems and Control Conference*. American Society of Mechanical Engineers, 2015, pp. V002T32A004–V002T32A004.

[8] L. Johannsmeier and S. Haddadin, "A hierarchical human-robot interaction-planning framework for task allocation in collaborative industrial assembly processes," *IEEE Robotics and Automation Letters*, vol. 2, no. 1, pp. 41–48, 2017.

[9] P. Tsarouchi, A.-S. Matthaiakis, S. Makris, and G. Chryssolouris, "On a human-robot collaboration in an assembly cell," *International Journal of Computer Integrated Manufacturing*, vol. 30, no. 6, pp. 580–589, 2017.

[10] M. C. Gombolay, R. Wilcox, and J. A. Shah, "Fast scheduling of multi-robot teams with temporospatial constraints." in *Robotics: Science and Systems*, 2013.

[11] Z. Shi, L. Wang, P. Liu, and L. Shi, "Minimizing completion time for order scheduling: Formulation and heuristic algorithm," *IEEE Transactions on Automation Science and Engineering*, 2015.

[12] Y. Li, K. P. Tee, R. Yan, W. L. Chan, and Y. Wu, "A framework of human–robot coordination based on game theory and policy iteration," *IEEE Transactions on Robotics*, vol. 32, no. 6, pp. 1408–1418, 2016.

[13] M. Schluse, M. Priggemeyer, L. Atorf, and J. Rossmann, "Experimentable digital twins - streamlining simulation-based systems engineering for industry 4.0," *IEEE Transactions on Industrial Informatics*, vol. 14, no. 4, pp. 1722–1731, April 2018.

[14] 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 *Humanoid Robots (Humanoids), 2013 13th IEEE-RAS International Conference on*. IEEE, 2013, pp. 499–506.

[15] C. Lenz, A. Sotzek, T. Röder, H. Radrich, A. Knoll, M. Huber, and S. Glasauer, "Human workflow analysis using 3d occupancy grid hand tracking in a human-robot collaboration scenario," in *Intelligent Robots and Systems (IROS), 2011 IEEE/RSJ International Conference on*. IEEE, 2011, pp. 3375–3380.

[16] C. Lenz, S. Nair, M. Rickert, A. Knoll, W. Rosel, J. Gast, A. Bannat, and F. Wallhoff, "Joint-action for humans and industrial robots for assembly tasks," in *Robot and Human Interactive Communication, 2008. RO-MAN 2008. The 17th IEEE International Symposium on*. IEEE, 2008, pp. 130–135.

[17] J. Kinugawa, A. Kanazawa, S. Arai, and K. Kosuge, "Adaptive task scheduling for an assembly task coworker robot based on incremental learning of human's motion patterns," *IEEE Robotics and Automation Letters*, vol. 2, no. 2, pp. 856–863, 2017.

[18] T. Ogata, S. Sugano, and J. Tani, "Open-end human–robot interaction from the dynamical systems perspective: mutual adaptation and incremental learning," *Advanced Robotics*, vol. 19, no. 6, pp. 651–670, 2005.

[19] W. Y. Kwon and I. H. Suh, "A temporal bayesian network with application to design of a proactive robotic assistant," in *Robotics and Automation (ICRA), 2012 IEEE International Conference on*. IEEE, 2012, pp. 3685–3690.

[20] V. Magnanimo, M. Saveriano, S. Rossi, and D. Lee, "A bayesian approach for task recognition and future human activity prediction," in *Robot and Human Interactive Communication, 2014 RO-MAN: The 23rd IEEE International Symposium on*. IEEE, 2014, pp. 726–731.

[21] J. Baraglia, M. Cakmak, Y. Nagai, R. P. Rao, and M. Asada, "Efficient human-robot collaboration: when should a robot take initiative?" *The International Journal of Robotics Research*, pp. 1–17, 2017.

[22] 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.

[23] D. Ron, Y. Singer, and N. Tishby, "The power of amnesia: Learning probabilistic automata with variable memory length," *Machine Learning*, vol. 25, no. 2-3, pp. 117–149, 1996.

[24] A. E. Raftery, "A model for high-order markov chains," *Journal of the Royal Statistical Society. Series B (Methodological)*, pp. 528–539, 1985.

[25] A. Raftery and S. Tavaré, "Estimation and modelling repeated patterns in high order markov chains with the mixture transition distribution model," *Applied Statistics*, pp. 179–199, 1994.

[26] W. K. Ching, E. S. Fung, and M. K. Ng, "Higher-order markov chain models for categorical data sequences," *Naval Research Logistics (NRL)*, vol. 51, no. 4, pp. 557–574, 2004.