Received: Added at production | Revised: Added at production | Accepted: Added at production

DOI: xxx/xxxx

ARTICLE TYPE

An Adaptive Learning Approach to Multivariate Time Forecasting in Industrial Processes

F. Miguelez[1,2] | **J. Doncel**[3] | **M. D. Ugarte**[1,2]

[1]Department of Statistics, Computer Science and Mathematics, Public University of Navarre, Navarre, Spain

[2]Institute for Advanced Materials and Mathematics (InaMat[2]), , Public University of Navarre, Navarre, Spain

[3]Department of Mathematics, University of the Basque Country, UPV/EHU, Leioa, Spain

Correspondence
Corresponding author F. Miguelez. Address: Edificio de los Magnolios, primera planta. Universidad Publica de Navarra. Campus de Arrosadia. 31006 Pamplona. Spain Spain
Email: fernando.miguelez@unavarra.es

Abstract

Industrial processes generate a massive amount of monitoring data that can be exploited to uncover hidden time losses in the system. This can be used to enhance the accuracy of maintenance policies and increase the effectiveness of the equipment. In this work, we propose a method for one-step probabilistic multivariate forecasting of time variables involved in a production process. The method is based on an Input-Output Hidden Markov Model (IO-HMM), in which the parameters of interest are the state transition probabilities and the parameters of the observations' joint density. The ultimate goal of the method is to predict operational process times in the near future, which enables the identification of hidden losses and the location of improvement areas in the process. The input stream in the IO-HMM model includes past values of the response variables and other process features, such as calendar variables, that can have an impact on the model's parameters. The discrete part of the IO-HMM models the operational mode of the process. The state transition probabilities are supposed to change over time and are updated using Bayesian principles. The continuous part of the IO-HMM models the joint density of the response variables. The estimate of the continuous model parameters is recursively computed through an adaptive algorithm that also admits a Bayesian interpretation. The adaptive algorithm allows for efficient updating of the current parameter estimates as soon as new information is available. We evaluate the method's performance using a real data set obtained from a company in a particular sector, and the results are compared with a collection of benchmark models.

KEYWORDS
Adaptive parameter estimates, Hidden Markov Model, Industrial processes, Probabilistic prediction

1 | INTRODUCTION

Machinery and equipment maintenance is the cornerstone of efficient and reliable production processes in industrial settings. With the increasing digitalization and automation of manufacturing lines, and the introduction of cyber-physical control platforms at the shop-floor level, the amount of available data has grown exponentially. This has led to the development of more sophisticated diagnostic and prognostic methodologies to identify and address small inefficiencies or hidden losses. In this context, industrial engineering experts are focusing on a proactive maintenance concept, which involves the early detection and correction of potential issues. Zwetsloot et al.[?] propose a method for early detection of changes in the frequency of out-of-control events in two signals, and apply it to study the health condition of escalators in some buildings in Hong Kong. The combination of Machine Learning and Deep Learning knowledge with operational data acquisition has led to the development of fault diagnosis methods for equipment in different working conditions. However, these methods typically focus on the degradation of mechanic components of highly specific equipment, but overlook external factors and other possible interactions that could affect the equipment's normal functioning, making them less effective in predicting overall system failures. Some benchmark examples on this matter are discussed by Yang and Zhong[?] . A proactive approach is generally more effective than a reactive one, which only addresses problems after they arise. Statistical methods, advanced analytics tools and machine learning algorithms have enabled the development of predictive maintenance models that try to predict equipment failures, preventing costly downtime

and production losses. The advent of continuous data flow in production processes is the basis of several applications that rely on real-time process monitoring, change point detection and the triggering of alerts in case of unusual trends. These methods belong to a process control concept known as Statistical Process Monitoring (SPM), and have proven to be a valuable resource for equipment health management. Woodall and Montgomery[?] provide a useful overview of techniques within this area. Unfortunately, since they are based on a mostly reactive approach, these methods still fail to predict the behaviour of the process in the near future and to anticipate far enough unplanned long stops caused by major breakdowns or micro stoppages caused by minor faults. This limitation has motivated the exploration of alternative techniques, such as Hidden Markov Models (HMMs). HMMs are flexible and mathematically robust, and have successfully modelled various applications, including speech[?] and handwriting recognition[?], electric consumption and generation forecasting[?], and DNA sequences analysis[?]. Nevertheless, the research community in the field of industrial process engineering agrees on being cautious about the straight utilization of these models due to the natural complexity and variability of industrial process data[?].

In this article we propose an innovative approach based on Input-Output Hidden Markov Models (IO-HMM) with adaptive learning for probabilistic multivariate forecasting of operational times in industrial processes. The methodology identifies hidden inefficiencies in production by estimating transitions between operational states and modelling the joint distribution of response variables. Through a dynamical parameter update based on the "Recursive Prediction Error" (RPE) technique[?], the model continuously improves its predictive capacity as new data is incorporated. The method is designed to be implemented in digital industrial management platforms and is applicable to a wide range of manufacturing sectors, while also being flexible enough to be customized to meet the specific needs of each company, including food processing, automotive, pharmaceutical and electronics, particularly in assembly line production, where minimizing downtime and optimizing workflow are critical for efficiency. This approach presents several key advantages for various stakeholders in an industrial setting. For maintenance managers, it enables early detection of equipment failures, facilitating preventive maintenance actions and reducing unplanned downtime. For production engineers, it enhances operational efficiency by identifying bottlenecks and optimizing production times. Industrial data analysts benefit from a robust analytical framework integrating probabilistic modelling with adaptive learning, allowing for better interpretation of process variability. Finally, for plant managers and executives, this approach facilitates data-driven decision-making by offering accurate predictions on equipment availability, performance, and quality.

A similar approach based on HMMs was proposed in Arpaia et al.[?] for detecting faulty conditions in fluid machinery. However, the aim of our model is not only the detection of the next operating mode but also the joint prediction of some operational time variables involved in a production process. From these operational times, one can deduce time losses, which reflect process inefficiencies that often remain undetected or overlooked, and some production effectiveness indices, which provide a reliable measure of the current performance of the process.

One of the challenges to overcome when dealing with HMMs is selecting the appropriate number of hidden states. This is especially meaningful in the context of equipment maintenance, since the hidden states are supposed to account for the general condition of the equipment under consideration. In Roblès et al.[?], authors examined the performance of different HMM topologies using well-known criteria such as the Bayesian Information Criterion, the Shannon Entropy and the Maximum Likelihood among others. The candidate models had different constraints over the transition matrix and different emission probability distributions but all of them were limited to four hidden states. However, this may not be sufficient for real-world applications where multiple intermediate levels may be present due to a variety of factors. Other authors use additional process' signals to determine the number of hidden states. Baruah and Chinnam[?] propose an experimental setting for diagnosing physical failure of drill bits and estimating remaining useful life using two highly correlated signals. In our approach, we address this issue by letting the data itself to determine the number of hidden states in a stage prior to the HMM modelling, adhering to general guidelines provided in Chinnam and Baruah[?].

As mentioned above, industrial processes are generally non-stationary in nature. One way to address this non-stationarity is to allow the parameter estimates to change over time incorporating explanatory variables in the parameter estimation procedure. Afzal and Al-Dabbagh[?] deal with a multi-signal process by considering an IO-HMM, an extension of the HMM that includes an input stream of variables that affect both the state transitions and the output densities[?]. In our approach, we adopt an IO-HMM model in which the parameter estimates depend on past values of the operational times and other process features, such as calendar variables (represented by work shifts) and production references. Following the suggestion of Baruah and Chinnam[?], the adaptive algorithm mentioned above ensures continuous parameter updating using the latest data.

The main novelty of this work is the handling of several process signals to identify potential faults in a challenging environment such as production processes. In particular, we focus on the analysis of multiple variables describing the production process carried out by a piece of equipment. Some of the variables are signals that characterize the health condition of the process,

and are used to establish the number of hidden states in the Markov chain of the model. Other variables are signals or process features that are considered to affect the response variables, and thus are used as explanatory variables that have an impact on the parameters of interest, i.e., the state transition probabilities and the parameters of the observations' joint density. The explanatory variables are called in the remainder of this work covariates. Further, the adaptive learning algorithm deployed in the continuous part of the IO-HMM is a multivariate extension of the algorithm presented in Alvarez et al.[?]

The rest of this paper is organised as follows. To provide better context, in Section 2 we define the operational times and indices relevant to this work, and describe the data of the case study that will be presented later. Section 3 outlines the IO-HMM and the methodology for parameter estimation and forecasting of response variables. Section 4 details the implementation process. The application to the real case study is introduced in Section 5. Finally, in Section 6 we discuss the conclusions.

2 | TIME LOSSES IN INDUSTRIAL PROCESSES AND APPLICATION DATA

In industrial settings, the production process is subject to inefficiencies that eventually assume the form of either output losses or time losses. When represented by time losses, they can be broadly classified into the following categories[?] :

1. Stand By Time (SBT): losses due to scheduled stops such as maintenance or cleaning
2. Down Time (DT): losses due to unexpected stops such as setup adjustments, failures, or supply outages
3. Performance Losses Time (PLT): losses due to low production speed and micro-stoppages
4. Quality Losses Time (QLT): losses associated with defective units and rework.

Note that each of these categories could further be subdivided based on the specific cause of the loss, although such a classification is typically customized according to the particular nature of the process under consideration. By taking the length of an observation period as a reference -hereinafter referred to as Opening Time or OT- one can derive different production times by successively subtracting each time loss, as illustrated in Figure 2.1 and definitions [2.1]. -

$$\begin{aligned} \mathtt{OT} - \mathtt{SBT} &= \text{Loading Time } (\mathtt{LT}) & \mathtt{OpT} - \mathtt{PLT} &= \text{Net Operating Time } (\mathtt{NOpT}) \\ \mathtt{LT} - \mathtt{DT} &= \text{Operating Time } (\mathtt{OpT}) & \mathtt{NOpT} - \mathtt{QLT} &= \text{Valuable Time } (\mathtt{VT}) \end{aligned} \qquad [2.1]$$

Moreover, the ratio between the production times can be used to define some well-known effectiveness indicators, which are enumerated in formulae [2.2]:

$$\begin{aligned} \frac{\mathtt{LT}}{\mathtt{OT}} &= \text{Loading Rate } (\mathtt{lo}) & \frac{\mathtt{NOpT}}{\mathtt{OpT}} &= \text{Performance Rate } (\mathtt{pf}) \\ \frac{\mathtt{OpT}}{\mathtt{LT}} &= \text{Availability Rate } (\mathtt{av}) & \frac{\mathtt{VT}}{\mathtt{NOpT}} &= \text{Quality Rate } (\mathtt{qu}) \end{aligned} \qquad [2.2]$$

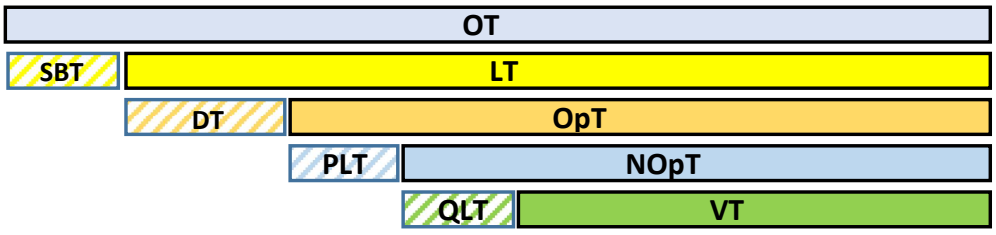

FIGURE 2.1 Production times and time losses classification from an observation period OT[?]

The Overall Equipment Effectiveness (OEE) is a widely-used index that weighs the actual capacity of equipment compared to its optimal capacity, and is defined as the product of the availability, performance and quality rates:

$$\texttt{oee} = \texttt{av} \times \texttt{pf} \times \texttt{qu} \quad [2.3]$$

The OEE is designed to trace the losses that are directly dependent on the equipment being used, while leaving out other losses that cannot be fixed by rearranging or repairing the equipment. Equivalently, the OEE can also be defined as

$$\texttt{oee} = \frac{\texttt{LT} - \texttt{DT} - \texttt{PLT} - \texttt{QLT}}{\texttt{LT}} = \frac{\texttt{VT}}{\texttt{LT}}$$

or

$$\texttt{oee} = \frac{\texttt{TU} - \texttt{DU}}{\texttt{ics} \times \texttt{LT}},$$

where ics is the ideal cycle speed (in cycles per time unit; a cycle, or unit, is a produced item), TU is the total number of units and DU the number of defective units. The last definition demonstrates that a 100% value in the OEE is obtained under optimal working conditions, that is, when the process has produced only flawless items at the ideal speed during the scheduled working hours. In Zammori et al.[?], the OEE is treated as a random variable and its distribution is used to assess the effectiveness of correction actions implemented in the maintenance strategy. In this study, different time losses are considered as independent beta random variables, but the independency assumption might be taken as unrealistic in real-life processes where, for example, a major failure is often preceded or followed by slower production speeds or by a higher number of rejected units. Our research will also explore whether the dependency between losses leads to a better predictive model by comparing the performances of the multivariate model and the respective univariate models.

To analyse these time losses and develop a predictive model for operational efficiency, we use a dataset collected from a real industrial process. The data comes from a company in a specific industrial sector, where a digital platform is integrated into the manufacturing line to capture and store real-time production data. The dataset comprises 1928 observations collected over four consecutive weeks, and include 35 variables related to the production process. Each observation corresponds to a specific period of operation, with key attributes listed in Table 2.1. A sample of two entries of the dataset is provided in Table 2.2, which illustrates the structure of the recorded observations and highlights the key variables involved in the analysis.

The motivation behind this research is to develop a probabilistic forecasting model capable of predicting key operational times in the near future. These predictions enable the identification of hidden inefficiencies and the calculation of process performance indicators, which are essential for proactive maintenance strategies, shop-floor decision-making and overall process optimization. However, industrial production dynamics are complex, driven by multiple interdependent factors, and often subject to both systematic patterns and stochastic variability. Therefore, traditional predictive models may struggle to capture these nuances, either because they oversimplify dependencies or fail to adapt to changing conditions. The multivariate IO-HMM provides a flexible framework for modelling the system's discrete operational modes using hidden states, incorporating explanatory variables to account for external factors affecting production performance, and dynamically adapting to new data through an adaptive learning algorithm that continuously updates model parameters. The real dataset will allow us to assess how effectively this approach achieves these objectives. It will also highlight the model's strengths while identifying potential areas for improvement, helping us reveal key aspects that require further attention.

3 | MODEL DESCRIPTION

Notation

Roman letters refer to scalar quantities or variables, lowercase bold letters denote vectors and uppercase bold letters denote matrices. Calligraphic letters refer to sets. $\mathbf{1}$ denotes a vector of 1's, $\mathbf{0}$ a vector or matrix of 0's, $\mathbf{I}$ is the identity matrix and $\cdot$ denotes the scalar product. Matrix or vector transposition is denoted by the superscript ^{T}.

TABLE 2.1 Key attributes in the real dataset.

Alias	Variable	Units/format
Production identifiers		
n	observation ID	integer
date	date	yyyy-mm-dd
start	timestamp	hh:mm:ss
shift	workshift	weekday-shift
pr.ord	production order ID	integer
Process parameters		
ics	ideal unit speed	units/minute
rcs	real unit speed (TU/LT)	units/minute
TU	total units	integer
DU	defective units	integer
TgU	target units (OpT×ics)	real
nstops	number of stops	integer
Time variables		
OT	Opening Time	minutes
SBT	Stand By Time	minutes
LT	Loading Time	minutes
DT	Downtime	minutes
OpT	Operating Time	minutes
PLT	Performance Losses Time	minutes
NOpT	Net Operating Time	minutes
QLT	Quality Losses Time	minutes
VT	Valuable Time	minutes
Indices		
lo	loading rate	$\in [0, 1]$
av	availability rate	$\in [0, 1]$
pf	performance rate	$\in [0, 1]$
qu	quality rate	$\in [0, 1]$
oee	OEE index	$\in [0, 1]$
Environmental variables		
hum	humidity	%
temp	temperature	°C

TABLE 2.2 An example of production data extracted from the real dataset.

n	date	start	shift	pr.ord	ics	TU	DU	TgU	OT	SBT	LT	rcs	lo
66	2022-10-10	13:50:24	Mo M	305	1.88	13	1	13.1	9.6	0	9.6	1.35	1
67	2022-10-10	14:00:00	Mo A	305	1.88	13	0	13.4	9.69	0	9.69	1.34	1

n	DT	OpT	av	PLT	NOpT	pf	QLT	VT	qu	oee	nstops	hum	temp
66	2.62	6.98	0.73	0.05	6.93	0.99	0.53	6.4	0.92	0.67	2	64.0	24.3
67	2.52	7.17	0.74	0.37	6.8	0.95	0	6.8	1	0.7	2	64.3	24.3

3.1 | Input-Ouput Hidden Markov Model

We model the production process as an IO-HMM, i.e., an HMM with an input stream of covariates. Figure 3.1 depicts the diagram of an IO-HMM. The main assumption in HMMs is that the process goes through K hidden states according to a state transition probability distribution. The hidden state of the n-th observation period is denoted by c_n and stands for the operational mode of the production process during that period. Each state gives rise to a different probability distribution of the continuous responses in the output stream, that are denoted by $\mathbf{y}_n$. The decision about the final number of hidden states will be discussed in Section 4.3.

The distinctive feature of an IO-HMM is the assumption that the model's parameters are affected by an input stream of covariates. This dependency allows the parameter estimates to change over time, in contrast with the traditional HMM, where the estimates are static after the training phase is completed. The covariates at n-th observation period comprehend prior known information about that period and are denoted by $\mathbf{x}_n$. The covariates affecting the state probabilities in the discrete part of the

model will be denoted by $\mathbf{z}_n$, while those that affect the responses' joint density will be denoted by $\mathbf{w}_n$, so that $\mathbf{x}_n = [\mathbf{z}_n\ \mathbf{w}_n]$. Both discrete and continuous processes of the model are described in sections 3.2 and 3.3 respectively.

In a simple framework, we can consider that the process's operational mode is exclusively determined by four levels of the OEE index: Optimal (>85%), Good (60-85%), Improvable (40-60%), and Poor (<40%). The specific operational mode during a given period remains unknown until the observation period ends, when we obtain access to the process information for that specific period, including the OEE score and the response variables $\mathbf{y}_n$. Based on this OEE, the period is categorized into one of the four levels. Using the prediction error $\mathbf{y}_n - \hat{\mathbf{y}}_n$, the last state c_n and the covariates $\mathbf{x}_{n+1}$, the model parameters are updated. Finally, with the covariates and the latest parameters, the value of the response vector $\hat{\mathbf{y}}_{n+1}$ is forecasted.

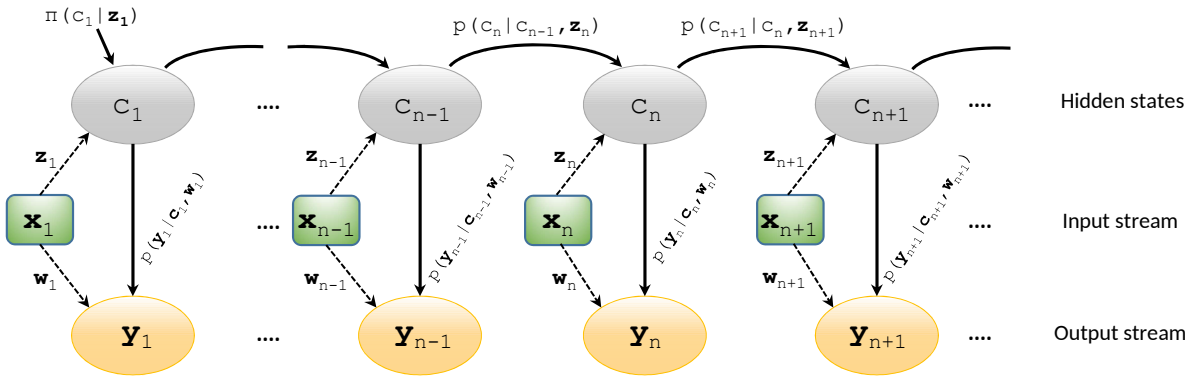

FIGURE 3.1 Diagram of an IO-HMM. Covariates $\mathbf{x}_n$ affect both discrete and continuous processes. Probabilities in the discrete process $\{c_n\}_{n\geq 1}$ are dependent on covariates $\mathbf{z}_n$ and probabilities in the continuous process $\{\mathbf{y}_n\}_{n\geq 1}$ are dependent on covariates $\mathbf{w}_n$.

3.2 | The discrete process

Assume that the discrete process $\{c_n\}_{n\geq 1}$ is a Markov chain with K different states, that is, $c_n \in \{1,\ldots,K\}$, $n \geq 1$. The probability distributions for the initial state and the transitions between states are dependent on the covariates $\mathbf{z}_n$. This vector $\mathbf{z}_n$ comprises d binary components that describes some features of the period, i.e., $\mathbf{z}_n \in \{0,1\}^d$, $n \geq 1$.

For a given $\mathbf{s} \in \{0,1\}^d$, we assume that the initial probabilities $\boldsymbol{\pi}^{(\mathbf{s})} = \mathbb{P}[c_1|\mathbf{z}_1 = \mathbf{s}]$ and the transition probabilities $\mathbf{p}_k^{(\mathbf{s})} = \mathbb{P}[c_n|c_{n-1} = k, \mathbf{z}_n = \mathbf{s}]$, $k = 1,\ldots,K$ follow Dirichlet prior distributions, that is,

$$\begin{aligned}\boldsymbol{\pi}^{(\mathbf{s})} &= \left[\pi_1^{(\mathbf{s})} \ldots \pi_K^{(\mathbf{s})}\right] \sim \text{Dirichlet}\left(\mathbf{a}^{(\mathbf{s})} = [a_1^{(\mathbf{s})} \ \ldots \ a_K^{(\mathbf{s})}]\right)\\ \mathbf{p}_k^{(\mathbf{s})} &= \left[p_{k1}^{(\mathbf{s})} \ \ldots \ p_{kK}^{(\mathbf{s})}\right] \sim \text{Dirichlet}\left(\boldsymbol{\alpha}_k^{(\mathbf{s})} = [\alpha_{k1}^{(\mathbf{s})} \ \ldots \ \alpha_{kK}^{(\mathbf{s})}]\right).\end{aligned} \tag{3.1}$$

It is well-known that the parameters of a Dirichlet distribution are recognized as pseudo-counts of the events represented by the random probabilities so that, for example, $a_k^{(\mathbf{s})}$ is the pseudo-count of sequences starting in state k with covariate $\mathbf{z}_1 = \mathbf{s}$, and $\alpha_{kj}^{(\mathbf{s})}$ is the pseudo-count of transitions from state k to state j when the covariates have the value $\mathbf{s}$. We will denote by

$$a_0^{(\mathbf{s})} \overset{def}{=} \sum_{k=1}^{K} a_k^{(\mathbf{s})} \quad \text{and} \quad \alpha_{k0}^{(\mathbf{s})} \overset{def}{=} \sum_{j=1}^{K} \alpha_{kj}^{(\mathbf{s})}$$

the concentration parameters of each distribution.

Conditioned to the last observed state, the next unobserved state is a random variable following a categorical distribution (or multinomial with one single trial) with parameters $\boldsymbol{\pi}^{(\mathbf{s})}$ if the sequence just begins, or $\mathbf{p}_k^{(\mathbf{s})}$ if the previous observation of the same sequence is in state k. Since a prior Dirichlet and a categorical likelihood are conjugate, the posterior distribution for the

parameters is also Dirichlet with revised pseudo-counts. Thus, when a new output measurement becomes available it is assigned to the closest state, say j, and the relevant pseudo-count is updated by increasing the parameter $a_j^{(\mathbf{s})}$ or $\alpha_{kj}^{(\mathbf{s})}$ by 1.

3.3 | The continuous process

The continuous process $\{\mathbf{y}_n\}_{n\geq 1}$ arises from a density function dependent on the state c_n and the covariates $\mathbf{w}_n$. To model these dependencies, we develop a multivariate extension of the model presented in Alvarez et al.[?], and split the conditional distribution of $\mathbf{y}_n|c_n, \mathbf{w}_n$ into two independent conditional distributions

$$\mathbf{y}_n|\mathbf{w}_n \sim N_m\left(\mathbf{u}_n\mathbf{H}_u, \mathbf{\Sigma}_u\right) \tag{3.2a}$$

$$\mathbf{y}_n|c_n \sim N_m(\mathbf{v}_n\mathbf{H}_v, \mathbf{\Sigma}_v), \tag{3.2b}$$

where m is the number of response variables; $\mathbf{H}_u, \mathbf{H}_v$ denote coefficient matrices; $\mathbf{\Sigma}_u, \mathbf{\Sigma}_v$ are covariance matrices; and $\mathbf{u}_n = [1\ \mathbf{w}_n]$, $\mathbf{v}_n = v(c_n)$ are the covariate vectors at the n-th step, with $v(\cdot)$ a function of the hidden state. In particular, we propose considering the conditional expectation $v(c_n) = \mathbb{E}\left[c_n|c_{n-1}\right]$, which, with the Dirichlet distribution assumptions, simplifies to the Dirichlet parameters in [3.1] normalized by their concentration parameters, i.e., $\mathbf{v}_n = \mathbf{a}/a_0$ or $\mathbf{v}_n = \boldsymbol{\alpha}_{c_{n-1}}/\alpha_{c_{n-1},0}$.

3.4 | The adaptive algorithm

We explain the adaptive algorithm for the case of the parameters in distribution [3.2a]; the same procedure is applied to the parameters in [3.2b]. Let $\lambda \in (0, 1]$ be a forgetting factor that accounts for the weight of past observations. As soon as a new sample $\mathbf{y}_n$ becomes available, the prior estimators $\mathbf{H}_{n-1}, \mathbf{\Sigma}_{n-1}$ are updated to $\mathbf{H}_n, \mathbf{\Sigma}_n$ (we omit the model subscripts in the parameters for clarity) through an adaptive algorithm described by the following equations:

$$\gamma_n = 1 + \lambda\gamma_{n-1} \tag{3.3a}$$

$$\mathbf{H}_n = \mathbf{H}_{n-1} + \frac{\mathbf{P}_{n-1}\mathbf{u}_n^T}{\lambda + \mathbf{u}_n\mathbf{P}_{n-1}\mathbf{u}_n^T}\left(\mathbf{y}_n - \mathbf{u}_n\mathbf{H}_{n-1}\right) \tag{3.3b}$$

$$\mathbf{\Sigma}_n = \mathbf{\Sigma}_{n-1} - \frac{1}{\gamma_n}\left[\mathbf{\Sigma}_{n-1} - \frac{\lambda\left(\mathbf{y}_n - \mathbf{u}_n\mathbf{H}_{n-1}\right)^T\left(\mathbf{y}_n - \mathbf{u}_n\mathbf{H}_{n-1}\right)}{\lambda + \mathbf{u}_n\mathbf{P}_{n-1}\mathbf{u}_n^T}\right] \tag{3.3c}$$

$$\mathbf{P}_n = \frac{1}{\lambda}\left(\mathbf{P}_{n-1} - \frac{\mathbf{P}_{n-1}\mathbf{u}_n^T\mathbf{u}_n\mathbf{P}_{n-1}}{\lambda + \mathbf{u}_n\mathbf{P}_{n-1}\mathbf{u}_n^T}\right) \tag{3.3d}$$

initialized with $\mathbf{H}_0 = \mathbf{0},\ \mathbf{\Sigma}_0 = \mathbf{0},\ \mathbf{P}_0 = \mathbf{I}$ and $\gamma_0 = 0$.γ_n is the total weight of the sample. Note that if $\lambda = 1$ then all the observations have the same weight and $\gamma_n = n$ is the sample size; if $\lambda < 1$, past data gradually loses influence over time while recent data has a greater impact. Equations [3.3a]-[3.3d] can be obtained by extending to the multivariate case the Maximum-Likelihood-based proof provided in Alvarez et al.[?], Theorem. 1. We refer to Appendix 6 for an alternative proof using a Bayesian approach.

3.5 | Forecasting

After the training step, each distribution [3.2a]-[3.2b] produces a forecast of the responses. Later, these forecasts are combined using a minimum-variance criterion to obtain the final prediction[?]. Once a new observation is available the update-prediction loop begins again. In particular, when the parameters are updated after the n-th observation is received we can write

$$\begin{aligned}
\mathbf{y}_{n+1,u} &= \mathbf{u}_{n+1}\mathbf{H}_u + \boldsymbol{\varepsilon}_{n+1}, \qquad \boldsymbol{\varepsilon}_{n+1} \sim N_m(\mathbf{0}, \mathbf{\Sigma}_u) \\
\mathbf{y}_{n+1,v} &= \mathbf{v}_{n+1}\mathbf{H}_v + \boldsymbol{\nu}_{n+1}, \qquad \boldsymbol{\nu}_{n+1} \sim N_m(\mathbf{0}, \mathbf{\Sigma}_v),
\end{aligned}$$

and define the weighted process

$$\mathbf{y}_{n+1} = \mathbf{y}_{n+1,u}\mathbf{D} + \mathbf{y}_{n+1,v}(\mathbf{I} - \mathbf{D}),$$

with $\mathbf{D} = \text{diag}\,(\delta_1, \ldots, \delta_m)$ a diagonal weight matrix to be determined. The mean and covariance of this process provide a multivariate forecast of the responses at time $(n+1)$ and an estimate of its accuracy, namely

$$\mathbb{E}_n\left[\mathbf{y}_{n+1}\right] \overset{def}{=} \hat{\mathbf{y}}_{n+1} = \mathbf{u}_{n+1}\mathbf{H}_u\mathbf{D} + \mathbf{v}_{n+1}\mathbf{H}_v(\mathbf{I} - \mathbf{D}) \quad [3.4a]$$

$$\mathbb{V}_n\left[\mathbf{y}_{n+1}\right] \overset{def}{=} \hat{\mathbf{\Sigma}}_{n+1} = \mathbf{D}\mathbf{\Sigma}_u\mathbf{D} + (\mathbf{I} - \mathbf{D})\mathbf{\Sigma}_v(\mathbf{I} - \mathbf{D}), \quad [3.4b]$$

where the subindex n in the expectation and variance operators denotes that they are applied given all the information available at time n. We note that finding $\mathbf{D}$ amounts to obtain separately the optimal weight δ_j for each response, $j = 1, \ldots, m$. This weight [?] is given by

$$\delta_j = \frac{\sigma_{v,j}^2}{\sigma_{u,j}^2 + \sigma_{v,j}^2}, \quad [3.5]$$

where $\sigma_{u,j}^2$ ($\sigma_{v,j}^2$) is the j-th diagonal element of $\mathbf{\Sigma}_u$ ($\mathbf{\Sigma}_v$).

4 | IMPLEMENTATION

4.1 | Data Segmentation

In an industrial setting, production processes are typically monitored at fixed intervals, and some time variables, such as `OT`, `SBT`, and `LT` in Figure 2.1, are often predetermined and treated as deterministic. This is because planned stops, such as maintenance or cleaning operations, follow predefined schedules, and measurement times are usually established in advance. However, in the dataset used for this study, which was described in Section 2, these variables are in fact random due to the specific method that employs the company that owns the data capture and management platform. Instead of using fixed measurement intervals, observations are recorded based on some operational events that occur randomly, making the observation times also inherently random, with the exception of one observation that is always recorded at the end of each shift. As a result, variables `OT`, `SBT`, and `LT` fluctuate depending on the data capture timing rather than being predefined, which introduces an additional source of variability in the model. If the process was monitored at predefined times and the variables `OT`, `SBT`, and `LT` were available a priori, the variability of the model would be reduced, and we would expect an improvement in the quality of the predictions.

To structure the data appropriately, each work shift is considered as a separate sequence of observation periods. That is, all observations recorded within a single shift form a continuous sequence in the model, and a new sequence begins when a shift change occurs. This segmentation ensures that the temporal dependencies within each shift are preserved, while taking into consideration the usual equipment adjustments that occur between shift changeovers. At the same time, we try to take advantage of the only measurement time that is known in advance, which is at the end of each shift.

4.2 | Variable selection

Covariate selection is a crucial step in the modelling process, as these variables must be meaningfully related to the response variables. In this context, a covariate is any observable characteristic of the process that potentially affects the dynamics of the target variables and, therefore, its inclusion in the model can enhance prediction accuracy. Some covariates influence the evolution of hidden states, affecting the initial and transition probabilities in the HMM, while others directly impact the joint distribution of output variables.

To ensure that the selected covariates genuinely add value to the model, dimensionality reduction methods such as Principal Component Analysis (PCA) can be employed when the set of available variables is too large. These techniques help identify combinations of variables that capture most of the data's variability without introducing redundancy. However, in industrial environments, the expertise of the production team is key to identifying relevant variables, so covariate selection can also be

also based on operational experience and prior correlation analysis with the response variables. A particular case of covariates includes past values of the response variables $\mathbf{y}_{n-1}, \ldots, \mathbf{y}_{n-q}$, as they may contain useful information about the future evolution of the process. The optimal number of lags to consider can be determined empirically through cross-validation by comparing the predictive performance of models with different autoregressive orders.

In addition to the covariates used for estimating the model parameters, it is also necessary to select a set of variables that enable the classification of observation periods into different operational states of the process. These classification variables are used in the clustering stage, and must be closely related to process efficiency and reflect its overall performance. Metrics such as availability rate, performance rate, OEE, and other key production indices are suitable candidates for this task. The selection of these variables can also be relied on data analysis techniques, or based on expert knowledge of the production process.

4.3 | Clustering

To identify the operational modes of the process, observation periods of the training set are grouped into $K \geq 2$ classes using an unsupervised classification technique based on the variables collected in the vector $\mathbf{t}_n$. Since the choice of classification technique is not the main objective of this work, for simplicity we applied the K-Means method at this stage in our case study, although other clustering approaches can certainly be explored; specifically, dynamic cluster merging and separation[?] techniques are particularly suited for non-stationary environments. Their incorporation into this type of model could enhance our understanding of the process dynamics, and is an interesting line of future research.

The optimal number of classes K, which corresponds to the number of hidden states in the Markov model, is not predefined but determined automatically. To minimize the need for manual model fitting, we establish this number as the minimum required for a goodness-of-fit (i.e., the between-groups-sum-of-squares divided by the total-sum-of-squares) threshold to be reached, adapting the number of hidden states to the actual complexity of the process and preventing both excessive segmentation and insufficient classification of the data. This will allow the application of the same method in settings with different conditions and equipment.

Once the classification is complete, each sequence in the training set is segmented into labelled intervals corresponding to the detected operational states. These states are later used to learn the parameters of the discrete part of the model, as described in Section 3.2, starting from non-informative Jeffreys' priors for the Dirichlet distributions [3.1], that is, $a_k = \alpha_{jk} = 1/2,\ \forall j, k$.

Let $\mathbf{o}_k$ be the centroid of the k-th class, $\mathbf{t}_n$ the classification variables, and c_n the class of the n-th observation. During the test phase, new observation periods are assigned to the closest centroid, that is

$$c_n = \underset{k \in \{1,\ldots,K\}}{\arg\min}\ d(\mathbf{o}_k, \mathbf{t}_n),$$

where $d(a, b)$ is a distance function such as Euclidean or Mahalanobis. Alternatively, the labels of the new observations can be obtained through a k-nearest neighbours classification scheme.

4.4 | Pseudo-code

Let $\mathcal{S}$ be the set of values of $\mathbf{z}_n$ and $\mathcal{K}$ the set of hidden states. We define the sets of parameters

$$\mathbf{\Pi} = \left\{ \mathbf{a_s} = [a_1^{(\mathbf{s})} \ \ldots \ a_K^{(\mathbf{s})}], \mathbf{A_s} = \left(\alpha_{kj}^{(\mathbf{s})}\right)_{k,j \in \mathcal{K}},\ \mathbf{s} \in \mathcal{S} \right\}$$

$$\mathbf{\Psi} = \left\{ \mathbf{H}_u^{(\mathbf{s})}, \mathbf{\Sigma}_u^{(\mathbf{s})}, \mathbf{H}_v^{(\mathbf{s})}, \mathbf{\Sigma}_v^{(\mathbf{s})},\ \mathbf{s} \in \mathcal{S} \right\}$$

$$\mathbf{\Omega} = \left\{ \mathbf{P}_u^{(\mathbf{s})}, \gamma_u^{(\mathbf{s})}, \mathbf{P}_v^{(\mathbf{s})}, \gamma_v^{(\mathbf{s})},\ \mathbf{s} \in \mathcal{S} \right\}$$

where

- $a_k^{(\mathbf{s})}$ is the count of sequences starting in state k with covariates $\mathbf{s}$,
- $\alpha_{kj}^{(\mathbf{s})}$ is the count of transitions from state k to state j for observations with covariates $\mathbf{s}$,

- $\mathbf{H}_{\bullet}^{(s)}, \mathbf{\Sigma}_{\bullet}^{(s)}$ are coefficient and covariance matrices respectively,
- $\mathbf{P}_{\bullet}^{(s)}, \gamma_{\bullet}^{(s)}$ are state matrices and discount factors respectively.

The pseudocode for the learning and forecasting methods is presented in Algorithms 1 and 2.

Algorithm 1 is responsible for updating the model parameters as new data become available. It follows an adaptive learning approach, ensuring that both discrete and continuous parts of the model are refined. The algorithm starts by constructing the necessary feature vectors $\mathbf{u}, \mathbf{v}$ for both distribution 3.2a- 3.2b in the continuous part of the model using the covariates and the normalized Dirichlet parameters, as suggested at the end of Section 3.3 (lines 2-6). In the continuous part, the model coefficients and covariance matrices are updated using the adaptive recursive estimation method presented in Section 3.4 (lines 7, 8). In the discrete part, the algorithm updates the prior state probabilities if a new sequence begins (line 11), or the transition probabilities otherwise (line 13). In any case, it is a simple update of the pseudo-counts that represent the parameters. The process is iterated over all observations (line 1), progressively refining the model's parameters to better reflect the dynamics of the system.

Algorithm 2 is in charge for predicting the next values of the response variables based on the most recent data and model estimates. First, the last observation is assigned to a hidden state based on the closest centroid obtained during the clustering step (line 1), and the centroid is updated dynamically (line 2). Next, feature vectors are created as in Algorithm 1 (lines 3-6), and the model estimates the response variables using both the hidden state-dependent process and the covariates-dependent process. The final forecast is a weighted combination of these two estimates (lines 8, 9), where the weights are determined based on the estimated variance of each process (line 7).

Figure 4.1 shows the block diagram of the adaptive algorithm for the model parameter estimates. At each step n, the algorithm receives the latest observation $\mathbf{y}_{n-1}$ and covariates $\mathbf{x}_n$, updating the model parameters based on the prediction error $(\mathbf{y}_{n-1} - \hat{\mathbf{y}}_{n-1})$. These updated parameters are used to generate the next forecast $\hat{\mathbf{y}}_n$ and the covariance matrix $\hat{\mathbf{\Sigma}}_n$, renewing the iterative learning-forecasting process.

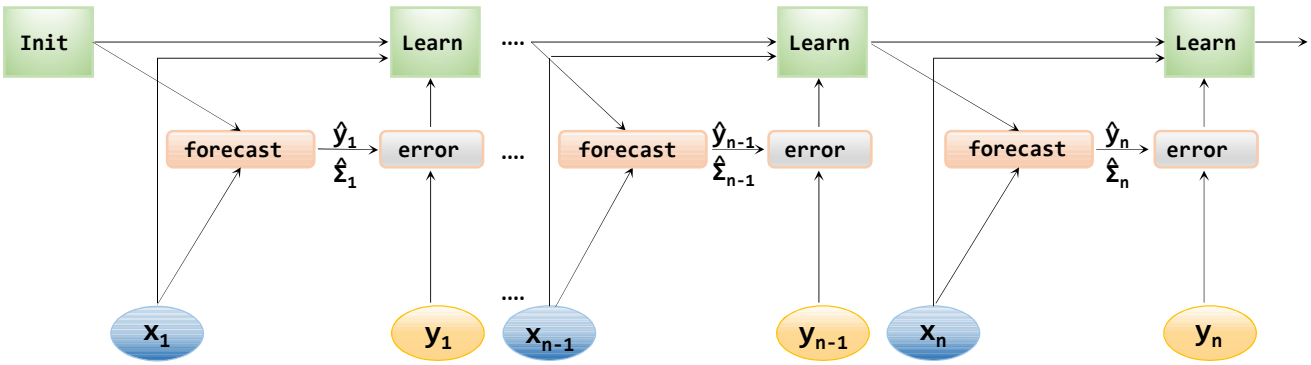

FIGURE 4.1 Block diagram of the adaptive algorithm for the parameter estimates. At the n-th step the error in the last prediction, $(\mathbf{y}_{n-1} - \hat{\mathbf{y}}_{n-1})$, and the new covariates, $\mathbf{x}_n$, feed the learning algorithm for updating the model parameters. The covariates and the latest parameters are then used to obtain the prediction.

4.5 | Evaluation

Two standard forecasting metrics are computed for the evaluation task -Mean Absolute Error (MAE) and Root Mean Squared Error (RMSE)- for each response variable. For a test set of L observations, these metrics are defined as

$$\text{MAE} = \frac{1}{L}\sum_{n=1}^{L} |y_n - \hat{y}_n|$$

Algorithm 1 Learning

Input:
$\boldsymbol{\Pi}$ ▷ Dirichlet parameters
$\boldsymbol{\Psi}$ ▷ model parameters
$\boldsymbol{\Omega}$ ▷ state parameters
λ_u, λ_v ▷ forgetting factors
$\mathbf{x}_n = [\mathbf{z}_n \ \ \mathbf{w}_n]$ ▷ covariates
$c_{n-1}, c_n, \mathbf{y}_n$ ▷ state labels, responses
Output:
$\boldsymbol{\Pi}, \ \boldsymbol{\Psi}, \ \boldsymbol{\Omega}$ ▷ updated parameters

for $n = 1, \ldots, N$ **do**
  $\mathbf{s} \leftarrow \mathbf{z}_n$
  $k \leftarrow c_{n-1}$
  $\mathbf{b} \leftarrow \begin{cases} \mathbf{a}^{(\mathbf{s})} & \text{if the sequence begins,} \\ \boldsymbol{\alpha}_k^{(\mathbf{s})} & \text{otherwise} \end{cases}$
  $\mathbf{u} \leftarrow [1 \ \ \mathbf{w}_n]$
  $\mathbf{v} \leftarrow \mathbf{b}/(\mathbf{b} \cdot \mathbf{1})$
  Update $\gamma_u^{(\mathbf{s})}, \mathbf{H}_u^{(\mathbf{s})}, \boldsymbol{\Sigma}_u^{(\mathbf{s})}, \mathbf{P}_u^{(\mathbf{s})}$ using equations [3.3a], [3.3b], [3.3c], [3.3d] respectively
  Update $\gamma_v^{(\mathbf{s})}, \mathbf{H}_v^{(\mathbf{s})}, \boldsymbol{\Sigma}_v^{(\mathbf{s})}, \mathbf{P}_v^{(\mathbf{s})}$ using equations [3.3a], [3.3b], [3.3c], [3.3d] respectively
  $j \leftarrow c_n$
  if the sequence begins **then**
    $a_j^{(\mathbf{s})} \leftarrow a_j^{(\mathbf{s})} + 1$
  else
    $\alpha_{kj}^{(\mathbf{s})} \leftarrow \alpha_{kj}^{(\mathbf{s})} + 1$
  end if
end for

$$\text{RMSE} = \sqrt{\frac{1}{L}\sum_{n=1}^{L}(y_n - \hat{y}_n)^2},$$

where y_n is the real value and $\hat{y}_n$ is the forecast. Further, to measure the probabilistic performance of the proposed model, we also compute for each response variable the coverage probability, which is defined as the proportion of real values that fall into the prediction interval:

$$covg = \frac{1}{L}\sum_{n=1}^{L}\mathbb{I}_{\{y_n \in [\hat{y}_n \pm 1.96\hat{\sigma}]\}},$$

where $\mathbb{I}$ is an indicator function and $\hat{\sigma}$ is the prediction error.

We compare the performance of the proposed model with different numbers of response lags, $q = 1, \ldots 5$, in the autoregressive component against the following benchmark models:

- The **persistence model**, which uses the last available observation to forecast the next one, that is $\hat{\mathbf{y}}_n = \mathbf{y}_{n-1}$. This is the baseline model, as all others are expected to produce better predictions.
- The **no-lags model**, which employs adaptive parameter learning without lag responses in the covariates (i.e., $q = 0$).
- The **Vector AutoRegressive model with exogenous variables** VARX(q), $q = 1, \ldots 5$. The general form of a VARX(q) model is[?]

$$\mathbf{y}_n = \boldsymbol{\phi}_0 + \sum_{j=1}^{q}\boldsymbol{\phi}_j \mathbf{y}_{n-j} + \boldsymbol{\beta}\mathbf{g}_n + \boldsymbol{\eta}_n,$$

Algorithm 2 Forecast

Input:
$\boldsymbol{\Psi}$ ▷ model parameters
$\{\mathbf{o}_1, \ldots, \mathbf{o}_K\}$ ▷ centroids
$\mathbf{t}_n$ ▷ classification variables
$\mathbf{x}_{n+1} = [\mathbf{z}_{n+1} \;\; \mathbf{w}_{n+1}]$ ▷ covariates
Output:
$\hat{\mathbf{y}}_{n+1}, \;\; \hat{\boldsymbol{\Sigma}}_{n+1}$ ▷ responses forecast, prediction error

$k \leftarrow \arg\min_{j \in \{1,\ldots,K\}} d(\mathbf{o}_j, \mathbf{t}_n)$
Update $\mathbf{o}_k$ including $\mathbf{t}_n$
$\mathbf{s} \leftarrow \mathbf{z}_{n+1}$
$\mathbf{b} \leftarrow \begin{cases} \mathbf{a}^{(\mathbf{s})} & \text{if the sequence begins} \\ \boldsymbol{\alpha}_k^{(\mathbf{s})} & \text{otherwise} \end{cases}$
$\mathbf{u} \leftarrow [1 \;\; \mathbf{w}_{n+1}]$
$\mathbf{v} \leftarrow \mathbf{b}/(\mathbf{b} \cdot \mathbf{1})$
Compute $\mathbf{D}$ using equation [3.5]
Compute $\hat{\mathbf{y}}_{n+1}$ using equation [3.4a]
Compute $\hat{\boldsymbol{\Sigma}}_{n+1}$ using equation [3.4b]

where $\boldsymbol{\phi}_j$ are VAR coefficient matrices, $\boldsymbol{\beta}$ is the coefficient matrix for the exogenous variables $\mathbf{g}_n$ and $\boldsymbol{\eta}_n$ is a sequence of iid random vectors of zero mean and positive-definite covariance matrix. To make a sound comparison, we include in $\mathbf{g}_n$ the same covariates as in $\mathbf{w}_n$ without the lagged responses, i.e., $\mathbf{w}_n = [\mathbf{g}_n \; \mathbf{y}_{n-1} \; \cdots \; \mathbf{y}_{n-q}]$. The VARX model does not account for the various operational modes of the process, and it is also a static model, meaning that the parameter estimates are not updated after the training stage. This makes this model adequate to assess the effect of both the inclusion of the discrete process described in Section 3.2 and the adaptive algorithm of Section 3.4.

- The respective **univariate models**, which we will use to check whether the multivariate approach takes advantage of the correlation structure between the responses.

5 | APPLICATION TO A REAL CASE STUDY

In this work, the model has been applied to real data provided by an industrial company. The availability of suitable public datasets for this type of research is very limited, mainly due to confidentiality agreements. Most open datasets related to industrial processes do not provide sufficiently detailed information on operational times, system states, or efficiency indicators, making it challenging to validate the model under realistic conditions. On the other hand, while simulated data could have been an alternative for testing the model in a controlled setting, generating realistic synthetic data for industrial processes is not straightforward. Simulating the dynamics of operational variables, hidden states, and interactions between different process factors requires assumptions that may not faithfully reflect the behaviour of a complex industrial system. For these reasons, the proposed model has been used to predict operational times in the production process of a company operating in a specific industrial sector. Due to confidentiality reasons, the exact industry of this company has been withheld. The data, described in Table 2.2, is supplied by the technological firm responsible for installing and maintaining the digital platform at the plant.

To properly feed the presented method, the data has been preprocessed, debugged and arranged using some well-known libraries in `Python` language[?] such as `pandas`[?] and `numpy`[?]. The core implementation of the procedure, as described in Section 4, has been developed in `R` language[?], and particularly the VARX models have been fitted using the MTS package[?].

The model's primary objective is to provide probabilistic forecasts of key operational times, thereby allowing production managers to anticipate inefficiencies and improve decision-making. For instance, if the model predicts an increase in performance losses (PLT) when switching between product types, experienced operators can be reassigned to these transitions to optimize setup times. Furthermore, forecasts of higher quality-related time losses (QLT) can prompt preemptive quality checks, enabling adjustments to machine settings to minimize defective units. Finally, if the model identifies patterns of increasing downtime (DT) near the end of long production runs, maintenance teams can schedule preventive maintenance activities before failure occurs.

The time variables OpT and NOpT are selected as response variables due to their strong correlation (0.83), which fully justifies the use of a multivariate model. According to the scheme presented in Figure 2.1 and the definitions [2.1], [2.2] and [2.3], predicting these variables allows for the computation of time losses due to low production speed and the performance index. Additionally, in scenarios where time measurements are equispaced and scheduled stops are predetermined (i.e., with deterministic OT, SBT and LT), these predictions also enable the estimation of time losses due to unexpected stops and the availability index.

The covariates in the discrete model ($\mathbf{z}_n$) include dummy variables that represent work shift levels, affecting the probability distribution of hidden states. The covariates in the continuous model ($\mathbf{w}_n$) incorporate the same dummy variables along with the ideal unit speed (ics) and two indicators marking the first observation of each shift and the first observation of each production order. Additionally, an autoregressive component is considered by including past response values, $\mathbf{y}_{n-1}, \ldots, \mathbf{y}_{n-q}$, within $\mathbf{w}_n$. For the classification step, the most relevant variables related to process health -including availability rate (av), performance rate (pf), overall equipment effectiveness (oee), opening time (OT), real unit speed (rcs) and total units produced (TU)- are used as classification criteria. These variables are collected in the vector $\mathbf{t}_n$.

A careful choice of forgetting factors λ_u, λ_v is essential to achieving a good trade-off between predictive accuracy and stability. One possible approach, not explored in this study but worth considering, is the use of dynamic forgetting factors[?], where λ_u and λ_v are adapted over time based on recent process variability. This could allow the model to better balance stability and responsiveness, particularly in non-stationary environments. Our initial experiments indicated that low values for both parameters produce very ill-conditioned state matrices $\mathbf{P}_n$, leading to unstable and unreliable predictions. This issue is mitigated when the values fall within the range of $[0.9, 1]$. From that point on, we found that for $0.9 \leq \lambda_u \leq 0.92$, the no-lag model consistently outperforms all other models across various metrics, and it remains the best in terms of RMSE and coverage for values $0.93 \leq \lambda_u \leq 0.95$. It is only when $\lambda_u \geq 0.96$ that lagged models begin to show improved performance, though the differences among lag-order models diminish progressively. Additionally, the MAE and RMSE metrics prove to be relatively stable for values of $\lambda_v \geq 0.9$, while coverage continues to improve at higher values. Therefore, we select the forgetting factors $\lambda_u = 0.99$, $\lambda_v = 0.95$ over a grid of values in the interval $[0.9, 1]$, although similar nearby values can yield comparable performance.

Six different models with q responses lags included in $\mathbf{w}_n$, $q = 0, 1, 2, 3, 4, 5$, are fitted using a Leave-One-Week-Out method, alternatively using three weeks of the dataset in the training step and the fourth week for prediction. MAE, RMSE and coverage are computed separately for each type of shift. We note that the average MAE and RMSE magnitudes are reasonable considering the responses sample quantiles and mean in the dataset, shown in Table 5.1. Figure 5.1 provides an insight into the distribution of the prediction error, MAE, RMSE and coverage across shifts for each output variable in this case study. The box-plot layouts suggest some key points:

(i) The proposed model outperforms all other models in terms of MAE. Since the MAE measures the mean absolute error, without giving greater weight to larger errors, this result suggests that the combination of an IO-HMM with adaptive learning captures the overall structure of the process better than other models, making it a suitable choice for optimizing processes where average performance is most important.

(ii) The improvement with respect to the RMSE metric is not as clear, being noticeable only for models with low lag-orders. This points to large errors occurring with a frequency comparable to that of other models, so adjustments should be considered if the objective is to detect uncommon events.

(iii) The prediction error of the proposed model is much smaller than the VAR(q) models, resulting in narrower confidence intervals, which also leads to a drop in coverage compared to the VAR(q) models. This suggests that the model underestimates the uncertainty in its predictions, which, linking with the previous point, can lead to large errors when unexpected deviations occur.

As can be observed, the coverage of the proposed model does not reach the nominal value of 95%, suggesting that it may be underestimating the variance of the prediction error. However, the extent of this underestimation is unclear. While coverage deviates from 95%, the comparable RMSE also indicates that extreme errors are not necessarily more frequent, which would be expected if uncertainty were severely underestimated. This phenomenon may be caused by several factors, such as limitations in the variance estimation method, a discrepancy between the assumed and actual error distributions, or an insufficient sample size for a reliable estimation. For example, some combinations of work shifts and hidden states may have too few cases to achieve a robust estimation of the variance of the prediction error.

TABLE 5.1 Responses sample mean and quantiles in the 4 weeks data set.

variable	min	Q1	median	mean	Q3	max
OpT	0	6.77	9.35	8.50	10.10	19.20
NOpT	0	6.77	9.35	8.44	10.08	19.02

Figure 5.2 presents a comparison between the multivariate model and its respective univariate counterparts. Despite the high correlation between response variables (0.83), which suggests that a multivariate approach should be beneficial, the results do not show a clear advantage of the multivariate model over the univariate ones. One possible explanation lies in the model's ability to properly estimate uncertainty. As previously discussed, the proposed model exhibits lower coverage than expected, indicating that it may be underestimating the variance of the prediction error. In contrast, univariate models, which estimate each response separately, may be less affected by variance underestimation and thus achieve comparable or even better performance in terms of predictive reliability. Another factor to consider is how well the multivariate structure captures cross-variable dependencies. While a high correlation suggests a strong relationship, it does not necessarily imply that a multivariate model will yield better predictions unless the dependencies are properly exploited. If the model's formulation does not fully capture the joint dynamics of the responses or the added complexity introduces estimation errors, the expected benefits of the multivariate approach may not materialize. Additionally, limited sample size or misspecification in the variance structure could further hinder its performance.

In summary, while the results of the proposed model applied to the case study seem promising, some areas for improvement have been recognized. Notably, the most critical issues appear to be addressing the underestimation of prediction error variance and improving the exploitation of dependencies between responses.

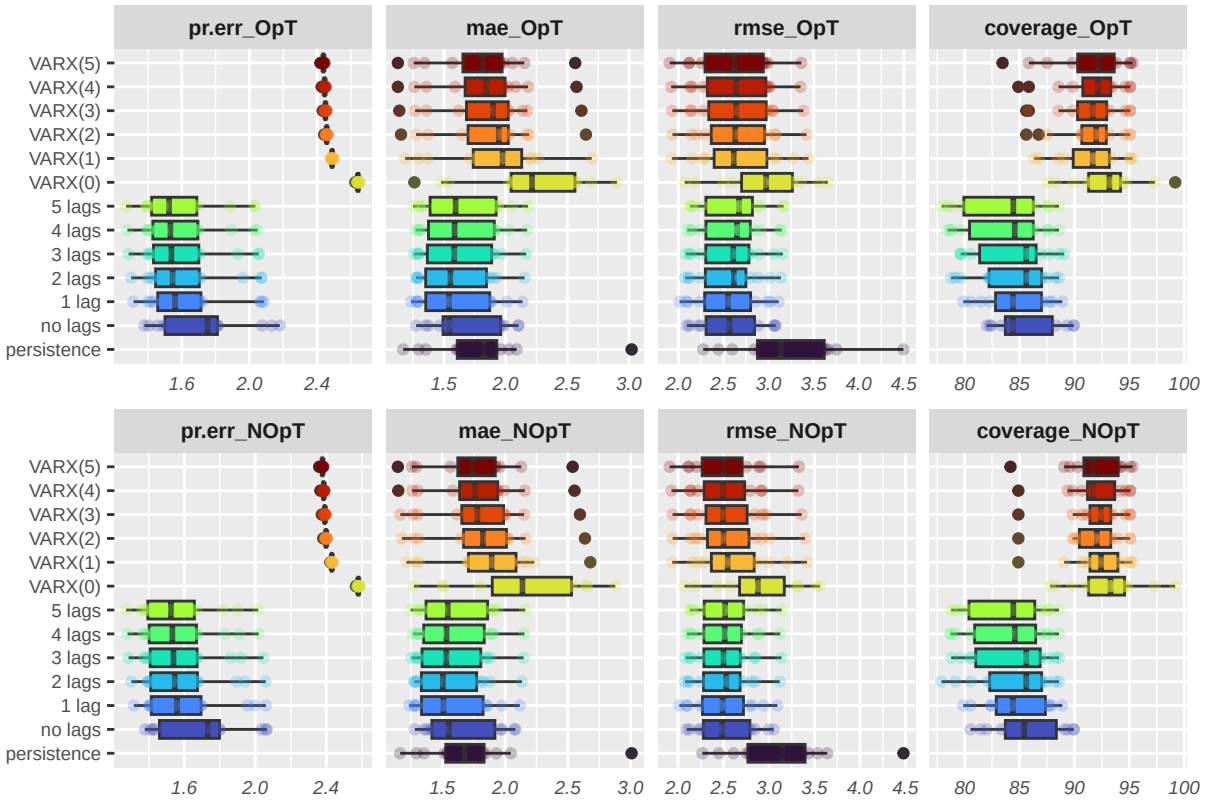

FIGURE 5.1 Boxplot of average prediction error, MAE, RMSE and coverage (by columns) across shifts for each output variable OpT and NOpT (by rows). The persistence model, no-lags model and VARX(q) models perform worse than the proposed model in terms of MAE, but VARX(q) models obtain comparable RMSE and better coverage due to the larger prediction error.

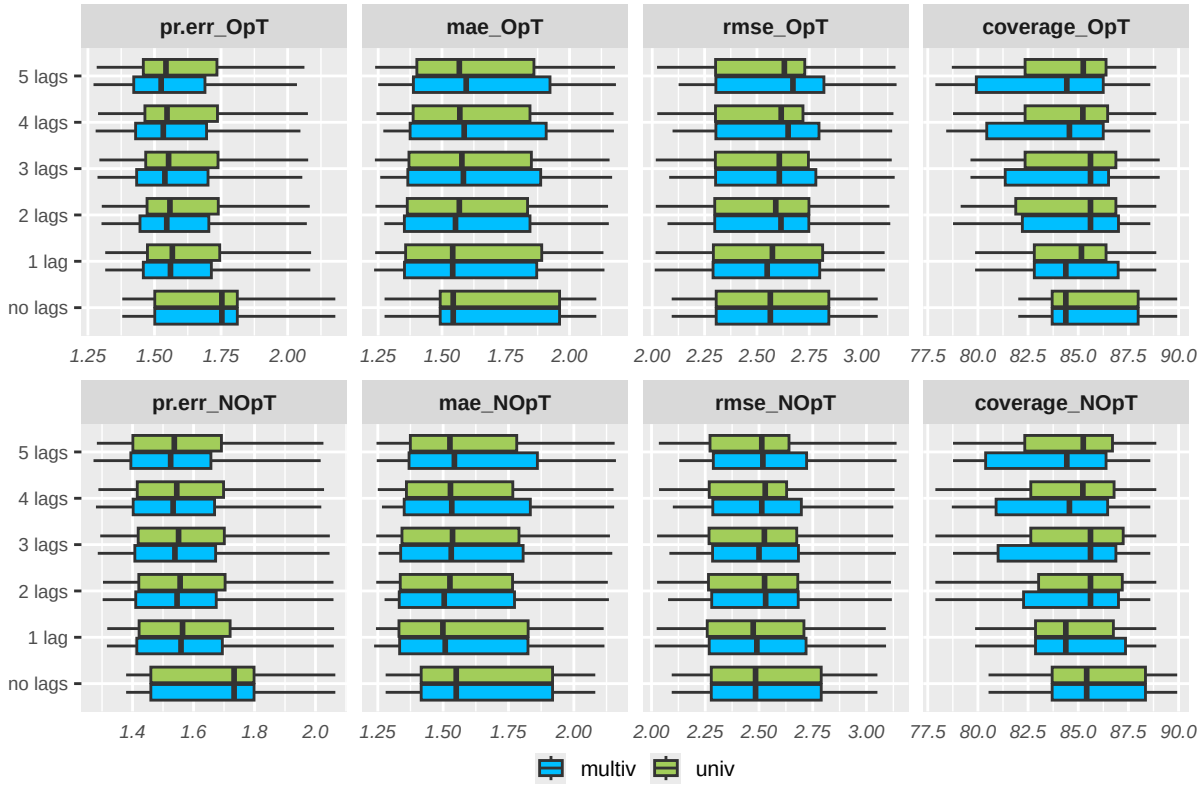

FIGURE 5.2 Comparison between multivariate and univariate models. Despite the high sample correlation between responses, there is no significant improvement in the performance of the multivariate model.

6 | CONCLUSIONS

This study introduces an adaptive learning approach for multivariate time forecasting in industrial processes using an Input-Output Hidden Markov Model (IO-HMM). By enabling probabilistic forecasting of operational times, the model provides a valuable tool for identifying inefficiencies and optimizing production workflows. The results demonstrate that the proposed method can be a helpful alternative against benchmark models, including univariate approaches and Vector AutoRegressive (VARX) models, by leveraging the multivariate nature of the data to capture dependencies between variables more effectively. The integration of an adaptive learning mechanism enables the model to dynamically adjust to new information, ensuring robust performance in non-stationary industrial environments. Through a Bayesian-inspired recursive update process, parameter estimates evolve continuously, making the approach particularly suited for real-time applications.

Beyond its predictive accuracy, the model effectively handles process variability by incorporating exogenous covariates such as production shifts, historical operational data, and other process-specific features. This adaptability, combined with a classification mechanism that automatically determines the number of hidden states, reduces the need for manual adjustments and enhances its applicability across different industrial settings. Designed to be highly scalable, the framework can be implemented in various production environments regardless of differences in equipment or manufacturing processes. Its flexibility allows seamless integration into digital industrial platforms, ensuring long-term usability even as process conditions evolve.

However, despite the theoretical advantages of a multivariate approach, the results indicate that univariate models can be equally competitive or, in some cases, even superior in predictive performance. Several factors could explain this outcome. First, estimating a multivariate model requires learning not only the relationship between each response variable and its covariates but also the dependencies between the responses. If these dependencies are unstable or vary significantly over time, a multivariate approach may not improve prediction accuracy. Moreover, the complexity of estimating a covariance structure can lead to overfitting, particularly when the sample size is not large enough to provide robust parameter estimates. In cases where the correlation between response variables does not provide substantial additional predictive power, a univariate model can achieve similar or better results with fewer parameters and less risk of overfitting. Another potential limitation arises from the way

prediction uncertainty is handled: while the multivariate model improves error estimates, it does not always achieve the expected coverage probability. This suggests that its variance structure might not be optimally calibrated, leading to an underestimation of the variance of prediction errors and narrow prediction intervals. Additionally, if the selected covariates already capture most of the information needed for prediction, the added complexity of modelling interdependencies may not translate into meaningful improvements.

The potential applications of this methodology extend beyond the specific case study, offering valuable insights for predictive maintenance by identifying early signs of equipment failure, enabling proactive strategies that minimize unplanned downtime. In production optimization, the model provides accurate forecasts that support real-time decision-making, helping to reduce bottlenecks and improve resource allocation. Its adaptability to real-time industrial monitoring makes it a reliable tool for continuous performance tracking and anomaly detection, further enhancing quality control by identifying patterns linked to production inefficiencies and allowing for early interventions to reduce waste. The approach can also be leveraged for energy and resource management, optimizing power consumption and scheduling by anticipating fluctuations in production efficiency.

This research highlights the value of combining IO-HMMs with adaptive learning techniques to develop a flexible and effective forecasting tool for complex industrial environments. By continuously updating its parameters and integrating multivariate dependencies, the model offers a powerful solution for improving decision-making in industrial processes. Future research could explore extensions to multi-step forecasting, integration with deep learning techniques, and broader scalability improvements to enhance its applicability across diverse sectors, including supply chain optimization and smart manufacturing systems. Additionally, further investigation into the conditions under which multivariate models provide a significant advantage over univariate approaches would help refine their practical use in industrial settings.

ACKNOWLEDGMENTS

This research was supported in part by the Government of Navarre under Project 0011-1365-2021-000085 and by the Department of Education of the Basque Government through the Consolidated Research Group MATHMODE (IT1456-22).

DATA AVAILABILITY STATEMENT

The data that support the findings of this study are available from the corresponding author upon reasonable request. The code to reproduce the results with anonymized data is available at the repository

https://github.com/spatialstatisticsupna/MTF-industrial-process.

□

APPENDIX

Proof of equations [3.3a]-[3.3d]

a) Following the directions of Rossi et al.[?], pp. 31–34, suppose a m-multivariate regression model with p predictor variables

$$\begin{cases} Y_1 = & X\beta_1 + \varepsilon_1 \\ Y_2 = & X\beta_2 + \varepsilon_2 \\ & \cdots \\ Y_m = & X\beta_m + \varepsilon_m \end{cases}$$

with errors correlated across equations. For the n-th observation in a random sample of size N,

$$\begin{bmatrix} y_{n1} \\ \vdots \\ y_{nm} \end{bmatrix} = \begin{bmatrix} \beta_1^T \\ \vdots \\ \beta_m^T \end{bmatrix} \begin{bmatrix} x_{n1} \\ \vdots \\ x_{np} \end{bmatrix} + \begin{bmatrix} \varepsilon_{n1} \\ \vdots \\ \varepsilon_{nm} \end{bmatrix}$$

$$\underset{m\times 1}{Y_n} = \underset{m\times p}{B^T}\underset{p\times 1}{X_n} + \underset{m\times 1}{\varepsilon_n} \qquad \text{with} \qquad \varepsilon_n \overset{iid}{\sim} \mathcal{N}_m(\mathbf{0}, \Lambda),\ n = 1, 2, \ldots, N$$

and gathering all

$$\begin{bmatrix} Y_1^T \\ \vdots \\ Y_N^T \end{bmatrix} = \begin{bmatrix} x_{11} & \cdots & x_{1p} \\ \vdots & \dots & \vdots \\ x_{N1} & \cdots & x_{Np} \end{bmatrix} \begin{bmatrix} \beta_1 & \cdots & \beta_m \end{bmatrix} + \begin{bmatrix} \varepsilon_1^T \\ \vdots \\ \varepsilon_N^T \end{bmatrix}$$

$$\underset{\uparrow \atop N\times m}{Y} = \underset{\uparrow \atop N\times pp\times m}{X}\ \underset{\uparrow}{B} + \underset{\uparrow \atop N\times m}{E}$$

For some $\Lambda_0 \in \mathcal{M}_{m\times m}$, $N_0 \in \mathbb{N}$, $\beta_0 \in \mathcal{M}_{mp\times 1}$, $V_0 \in \mathcal{M}_{p\times p}$, the natural conjugate priors for the parameters in the multivariate regression model can be taken as

$$\begin{aligned} \mathbb{P}[\Lambda, B] &= \mathbb{P}[B|\Lambda]\,\mathbb{P}[\Lambda] \\ \Lambda &\sim \mathcal{W}^{-1}(N_0\Lambda_0, m, N_0 + m + 1) \\ \beta|\Lambda &\sim \mathcal{N}_{mp}\left(\beta_0, \Lambda \otimes V_0^{-1}\right) \end{aligned}$$

where $\mathcal{W}^{-1}$ denotes a inverted Wishart distribution, $\otimes$ is the Kronecker product and $\beta = \mathrm{vec}(B)$ (vectorization). The prior means are $\mathbb{E}[B|\Lambda] = B_0$ and $\mathbb{E}[\Lambda] = \Lambda_0$.

Given these priors and the random sample, the posterior joint density for the parameters can be decomposed into the product of the following densities

$$\begin{aligned} \Lambda|Y, X &\sim \mathcal{W}^{-1}\left(N_0\Lambda_0 + N\tilde{S}, m, N_0 + N + m + 1\right) \\ \beta|\Lambda, Y, X &\sim \mathcal{N}_{mp}\left(\tilde{\beta}, \Lambda \otimes \left(X^TX + V_0\right)^{-1}\right) \end{aligned}$$

where

$$\begin{aligned} \tilde{\beta} = \mathrm{vec}\left(\tilde{B}\right), \quad \tilde{B} &= B_0 + \left(X^TX + V_0\right)^{-1} X^T\left(Y - XB_0\right) \\ N\tilde{S} &= \left(Y - X\tilde{B}\right)^T\left(Y - X\tilde{B}\right) + \left(\tilde{B} - B_0\right)^T V_0\left(\tilde{B} - B_0\right). \end{aligned} \tag{1a}$$

Hence, the posterior means are

$$\mathbb{E}[B|\Lambda] = \tilde{B} = B_0 + \left(X^TX + V_0\right)^{-1} X^T\left(Y - XB_0\right) \tag{2a}$$

$$\begin{aligned} \mathbb{E}[\Lambda|Y] &= (N_0 \underset{\uparrow \atop \text{prior mean}}{\Lambda_0} + N\tilde{S})/(N_0 + N) \\ &= \Lambda_0 - \frac{N\left(\Lambda_0 - \tilde{S}\right)}{N_0 + N}. \end{aligned} \tag{2b}$$

b) To make the above results fit in with our case let us define

$$\begin{aligned} Y &= \mathbf{y}_n \ \text{(new responses, row vector } 1\times m) \\ X &= \mathbf{u}_n \ \text{(new predictors, row vector } 1\times p) \\ \tilde{B} &= \mathbf{H}_n \ \text{(posterior mean, } p\times m) \\ B_0 &= \mathbf{H}_{n-1} \ \text{(prior mean, } p\times m) \\ V_0 &= \lambda\mathbf{P}_{n-1}^{-1} \ \text{(prior state matrix, } p\times p) \\ (X^TX + V_0)^{-1} &= \mathbf{P}_n \ \text{(posterior state matrix, } p\times p) \\ \gamma_n &= 1 + \lambda + \cdots\lambda^{n-1} \ \text{(scalar)} \end{aligned}$$

Equation [3.3a] is obvious given that $\gamma_n = 1 + \lambda\left(1 + \lambda + \cdots\lambda^{n-2}\right) = 1 + \lambda\gamma_{n-1}$. From the above definition of the posterior state matrix $\mathbf{P}_n$ and using the Sherman-Morrison formula

$$\begin{aligned} \mathbf{P}_n &= \left(X^TX + V_0\right)^{-1} \\ &= \left(\mathbf{u}_n^T\mathbf{u}_n + \lambda\mathbf{P}_{n-1}^{-1}\right)^{-1} \end{aligned}$$

$$= \left(\lambda \mathbf{P}_{n-1}^{-1}\right)^{-1} - \frac{\left(\lambda \mathbf{P}_{n-1}^{-1}\right)^{-1} \mathbf{u}_n^T \mathbf{u}_n \left(\lambda \mathbf{P}_{n-1}^{-1}\right)^{-1}}{1 + \mathbf{u}_n \left(\lambda \mathbf{P}_{n-1}^{-1}\right)^{-1} \mathbf{u}_n^T}$$
$$= \frac{1}{\lambda}\left(\mathbf{P}_{n-1} - \frac{\mathbf{P}_{n-1}\mathbf{u}_n^T \mathbf{u}_n \mathbf{P}_{n-1}}{\lambda + \mathbf{u}_n \mathbf{P}_{n-1}\mathbf{u}_n^T}\right)$$

we obtain equation [3.3d]. For the posterior mean $\tilde{B} = \mathbf{H}_n$, using expression [2a] we have

$$\begin{aligned}
\mathbf{H}_n = \tilde{B} &= B_0 + \left(X^T X + V_0\right)^{-1} X^T (Y - XB_0) \\
&= \mathbf{H}_{n-1} + \mathbf{P}_n \mathbf{u}_n^T (\mathbf{y}_n - \mathbf{u}_n \mathbf{H}_{n-1}) \\
&= \mathbf{H}_{n-1} + \frac{1}{\lambda}\left(\mathbf{P}_{n-1} - \frac{\mathbf{P}_{n-1}\mathbf{u}_n^T \mathbf{u}_n \mathbf{P}_{n-1}}{\lambda + \mathbf{u}_n \mathbf{P}_{n-1}\mathbf{u}_n^T}\right) \mathbf{u}_n^T (\mathbf{y}_n - \mathbf{u}_n \mathbf{H}_{n-1}) \\
&= \mathbf{H}_{n-1} + \frac{1}{\lambda} \frac{\mathbf{P}_{n-1}\mathbf{u}_n^T \left(\lambda + \mathbf{u}_n \mathbf{P}_{n-1}\mathbf{u}_n^T\right) - \mathbf{P}_{n-1}\mathbf{u}_n^T \mathbf{u}_n \mathbf{P}_{n-1}\mathbf{u}_n^T}{\lambda + \mathbf{u}_n \mathbf{P}_{n-1}\mathbf{u}_n^T} (\mathbf{y}_n - \mathbf{u}_n \mathbf{H}_{n-1}) \\
&= \mathbf{H}_{n-1} + \frac{\mathbf{P}_{n-1}\mathbf{u}_n^T}{\lambda + \mathbf{u}_n \mathbf{P}_{n-1}\mathbf{u}_n^T} (\mathbf{y}_n - \mathbf{u}_n \mathbf{H}_{n-1}),
\end{aligned}$$

yielding equation [3.3b]. On the other hand, from the definition [1a] we get

$$\begin{aligned}
N\tilde{S} &= \left(Y - X\tilde{B}\right)^T \left(Y - X\tilde{B}\right) + \left(\tilde{B} - B_0\right)^T V_0 \left(\tilde{B} - B_0\right) \\
&= (\mathbf{y}_n - \mathbf{u}_n \mathbf{H}_n)^T (\mathbf{y}_n - \mathbf{u}_n \mathbf{H}_n) + (\mathbf{H}_n - \mathbf{H}_{n-1})^T \left(\lambda \mathbf{P}_{n-1}^{-1}\right) (\mathbf{H}_n - \mathbf{H}_{n-1}) \\
&= \left[\mathbf{y}_n - \mathbf{u}_n \left(\mathbf{H}_{n-1} + \frac{\mathbf{P}_{n-1}\mathbf{u}_n^T}{\lambda + \mathbf{u}_n \mathbf{P}_{n-1}\mathbf{u}_n^T} (\mathbf{y}_n - \mathbf{u}_n \mathbf{H}_{n-1})\right)\right]^T \left[\mathbf{y}_n - \mathbf{u}_n \left(\mathbf{H}_{n-1} + \frac{\mathbf{P}_{n-1}\mathbf{u}_n^T}{\lambda + \mathbf{u}_n \mathbf{P}_{n-1}\mathbf{u}_n^T} (\mathbf{y}_n - \mathbf{u}_n \mathbf{H}_{n-1})\right)\right] + \\
&\quad \left[\frac{\mathbf{P}_{n-1}\mathbf{u}_n^T}{\lambda + \mathbf{u}_n \mathbf{P}_{n-1}\mathbf{u}_n^T} (\mathbf{y}_n - \mathbf{u}_n \mathbf{H}_{n-1})\right]^T \left(\lambda \mathbf{P}_{n-1}^{-1}\right) \left[\frac{\mathbf{P}_{n-1}\mathbf{u}_n^T}{\lambda + \mathbf{u}_n \mathbf{P}_{n-1}\mathbf{u}_n^T} (\mathbf{y}_n - \mathbf{u}_n \mathbf{H}_{n-1})\right] \\
&= \left[(\mathbf{y}_n - \mathbf{u}_n \mathbf{H}_{n-1}) - \frac{\mathbf{u}_n \mathbf{P}_{n-1}\mathbf{u}_n^T}{\lambda + \mathbf{u}_n \mathbf{P}_{n-1}\mathbf{u}_n^T} (\mathbf{y}_n - \mathbf{u}_n \mathbf{H}_{n-1})\right]^T \left[(\mathbf{y}_n - \mathbf{u}_n \mathbf{H}_{n-1}) - \frac{\mathbf{u}_n \mathbf{P}_{n-1}\mathbf{u}_n^T}{\lambda + \mathbf{u}_n \mathbf{P}_{n-1}\mathbf{u}_n^T} (\mathbf{y}_n - \mathbf{u}_n \mathbf{H}_{n-1})\right] + \\
&\quad \lambda \frac{\mathbf{u}_n \mathbf{P}_{n-1}\mathbf{u}_n^T}{\left(\lambda + \mathbf{u}_n \mathbf{P}_{n-1}\mathbf{u}_n^T\right)^2} (\mathbf{y}_n - \mathbf{u}_n \mathbf{H}_{n-1})^T (\mathbf{y}_n - \mathbf{u}_n \mathbf{H}_{n-1}) \\
&= \left[\frac{\lambda (\mathbf{y}_n - \mathbf{u}_n \mathbf{H}_{n-1})}{\lambda + \mathbf{u}_n \mathbf{P}_{n-1}\mathbf{u}_n^T}\right]^T \left[\frac{\lambda (\mathbf{y}_n - \mathbf{u}_n \mathbf{H}_{n-1})}{\lambda + \mathbf{u}_n \mathbf{P}_{n-1}\mathbf{u}_n^T}\right] + \lambda \frac{\mathbf{u}_n \mathbf{P}_{n-1}\mathbf{u}_n^T}{\left(\lambda + \mathbf{u}_n \mathbf{P}_{n-1}\mathbf{u}_n^T\right)^2} (\mathbf{y}_n - \mathbf{u}_n \mathbf{H}_{n-1})^T (\mathbf{y}_n - \mathbf{u}_n \mathbf{H}_{n-1}) \\
&= \frac{\lambda (\mathbf{y}_n - \mathbf{u}_n \mathbf{H}_{n-1})^T (\mathbf{y}_n - \mathbf{u}_n \mathbf{H}_{n-1})}{\lambda + \mathbf{u}_n \mathbf{P}_{n-1}\mathbf{u}_n^T}.
\end{aligned}$$

Now, by choosing $N_0 = \lambda\gamma_{n-1}$, $N = 1$ and

$$\begin{aligned}
\Lambda_0 &= \boldsymbol{\Sigma}_{n-1} \quad \text{(prior mean variance, } m \times m\text{)} \\
\mathbb{E}[\Lambda | Y] &= \boldsymbol{\Sigma}_n \quad \text{(posterior mean variance, } m \times m\text{)}
\end{aligned}$$

and using expression [2b]

$$\begin{aligned}
\boldsymbol{\Sigma}_n = \mathbb{E}[\Lambda | Y] &= \Lambda_0 - \frac{N\left(\Lambda_0 - \tilde{S}\right)}{N_0 + N} \\
&= \boldsymbol{\Sigma}_{n-1} - \frac{1}{1 + \lambda\gamma_{n-1}}\left[\boldsymbol{\Sigma}_{n-1} - \frac{\lambda (\mathbf{y}_n - \mathbf{u}_n \mathbf{H}_{n-1})^T (\mathbf{y}_n - \mathbf{u}_n \mathbf{H}_{n-1})}{\lambda + \mathbf{u}_n \mathbf{P}_{n-1}\mathbf{u}_n^T}\right] \\
&= \boldsymbol{\Sigma}_{n-1} - \frac{1}{\gamma_n}\left[\boldsymbol{\Sigma}_{n-1} - \frac{\lambda (\mathbf{y}_n - \mathbf{u}_n \mathbf{H}_{n-1})^T (\mathbf{y}_n - \mathbf{u}_n \mathbf{H}_{n-1})}{\lambda + \mathbf{u}_n \mathbf{P}_{n-1}\mathbf{u}_n^T}\right]
\end{aligned}$$

follows equation [3.3c].