

Automatic State Matching Gaussian Process Ensemble for Wood Planer Control

Jean-Thomas Sexton^{*,**} Michael Morin^{*,***}
Rémi Georges^{****} Foroogh Abasian^{****}
Jonathan Gaudreault^{*,**}

** FORAC Research Consortium*

*** Department of Computer Science and Software Engineering*

**** Department of Operations and Decision Systems*

Université Laval, Québec, QC, Canada

*(e-mail: jtsex@ulaval.ca, jonathan.gaudreault@ift.ulaval.ca,
michael.morin@osd.ulaval.ca)*

***** FPInnovations, Québec, QC, Canada*

*(e-mail: remi.georges@fpinnovations.ca,
foroogh.abasian@fpinnovations.ca)*

Abstract: Wood planers are high speed sophisticated lumber finishing machines that are difficult to operate and for which the available data shows complex, non-linear patterns. We present a machine learning approach to build a control loop for an industrial wood planer. In order to predict the thickness of the outgoing boards with better accuracy than the industry standard whilst allowing dynamic planer adjustments, we use an ensemble of Gaussian Processes with a specialized weighting scheme we call **Automatic State Matching**. It reduces the prediction error by 39% compared to current industrial practice.

Keywords: Smart manufacturing, Machine learning, Production control, Gaussian process, Industry 4.0

1. INTRODUCTION

Canadian forest-product companies face serious challenges due to current trends towards globalization and worker shortages. Experienced planer operators are increasingly difficult to find. Although planers are highly flexible machines, adjustments are mostly performed only when a problem is detected and the planer is idle. This leads to the production of boards with less desirable dimensions. The infrequency of adjustments is mostly due to the difficulty of finding experienced operators, and the high cost of mistakes. More frequent and dynamic adjustments could keep the outgoing boards closer to the desired dimensions which would generate additional profits, and automated adjustments could alleviate the issue of the lack of workforce. Following this idea, we address the following research question: Would it be possible to design a machine learning based control loop in order to automate a high-speed sophisticated wood planer?

As an answer to that question, we propose a novel algorithm which will be used in a control loop to adjust the planer in real time. Historically, the main tool used in industrial control loops is PID control (Johnson and Moradi, 2005). In recent years, trends towards data collection have facilitated the rise of predictive machine learning models in order to provide proactive control (Camacho and Alba, 2013) and process control in production planning (Usuga Cadavid et al., 2020). Other related approaches are reinforcement learning (Barto, 1994) where performance is improved through trial-and-error, and deep

reinforcement learning which dramatically extends the level of complexity in process dynamics that can be learned compared to traditional reinforcement learning (Lillicrap et al., 2015). Our partners from the industry reported that the dynamics of the wood planer are too complex for optimal PID control. In addition, due to the high financial costs of mistakes (and the absence of a realistic simulator capturing the high variability/instability of the process), we cannot experiment directly on the planer which would be needed for reinforcement learning (deep or otherwise). Thus, we focus on model predictive control. We propose to model the system as an ensemble of Gaussian Processes (one for each previous production batch). Weights are dynamically computed to match in real time the current batch to the previous batches. This allows dealing with the high variability of the process.

The paper is organized as follows: In Section 2, we review the necessary concepts and challenges related to the task of building a control loop for a wood planer, as well as the available industrial data. Section 3 details the proposed approach. In Section 4, we explain the experiments we performed to evaluate the proposed approach. The results of those experiments are discussed in Section 5. We conclude in Section 6.

2. PRELIMINARY CONCEPTS AND CHALLENGES

A lumber planer is used to trim rough boards (lumbers) to the desired thickness and/or width in order to be sold on the market. The value/quality of a lumber depends on

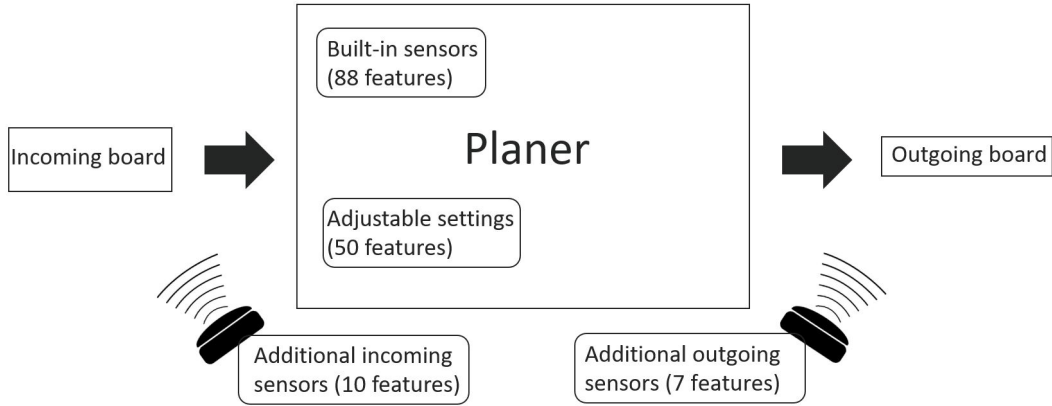


Fig. 1. Relevant components of the planer and number of gathered features for each component

how close its dimensions are to industry standards with respect to the particular desired size.

Figure 1 shows a planer diagram with the relevant components along with the number of data points (features) we consider for each component. There are 50 control settings which can be adjusted by the operators (e.g., the position of the knives). The built-in sensors of the planer provide 88 data points during the processing of a board (e.g., the temperature inside the planer). External sensors provide 10 more informative measures about the board before it is processed and 7 afterwards (e.g., the temperature of the board before and after being processed).

When the machine is idle for more than 60 seconds, it is highly likely that a change has occurred (mechanical or parameters adjustments by the operators or, changes in the type of wood that is being processed). Consequently, sequential data acquired without any interruptions longer than 60 seconds is defined as a **batch**.

Figure 2 shows the thicknesses of boards for two consecutive batches where the adjustable settings are the same. We can see that boards are, on average, thicker in the second batch, despite no change in adjustable settings being recorded in the database.

An important adjustable/controllable setting of the planer is called the “OFFSET”. It can be used to move the knives closer or further to the wood in order to adjust the expected thickness. However, this is infrequently done as there is a lot of noise in the data and operators are not confident enough to do it in real time, neither the mills are confident enough to implement a traditional feedback loop for the same reason.

3. PROPOSED APPROACH: SMART PLANER CONTROL WITH GAUSSIAN PROCESSES

Our data was collected in a sawmill in Quebec, Canada. We focused on a single product (one of the most available standard-size lumber products of North America called “2×4×96”) for which we acquired a total of 155 data points each time a lumber is processed (148 before the board is processed and 7 afterwards). Preliminary analysis showed that the distribution of the boards dimensions coming out of the planer can be dramatically different from

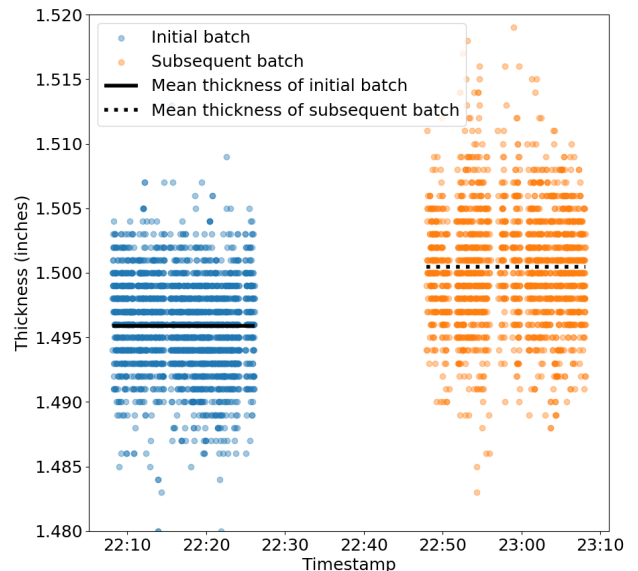


Fig. 2. Thickness of “2×4×96” boards for two consecutive production batches with identical adjustable settings. Outliers were cut out to make the figure more readable.

one batch to another. As an example, Figure 2 reports board thickness for two consecutive batches. Although no settings were changed in the wood planer between those batches, we can yet observe two very different processes. The non-stationary nature of the data could be due to the fact that some important variables are unknown, e.g., the type of wood, the humidity in the wood, the wear and tear of the planer knives and changes in mechanical parameters. This leads us to believe that using a simple machine learning model trained on all available data without taking into account its non-stationary nature (and the concept of batches) could be difficult.

Therefore, we propose an approach based on Gaussian Processes (GPs). Gaussian processes have found a wide range of applications in systems control (Kocijan, 2016). Due to the amount of data, the non-stationary nature of

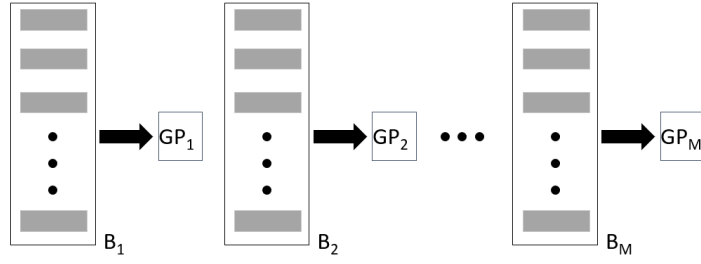


Fig. 3. Overview of the training procedure. For each past batch B_1, B_2, \dots, B_M , the information from each board, including related features from the planer, are used to train a Gaussian Process GP_1, GP_2, \dots, GP_M .

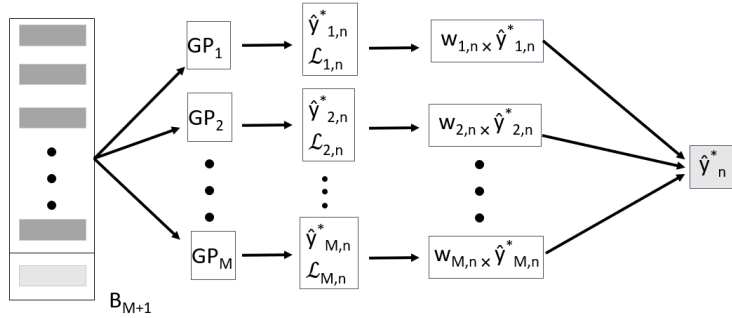


Fig. 4. Overview of the computations for the prediction of the thickness of the n^{th} board of the current batch B_{M+1} . We use the history of the $n - 1$ previous boards from the current batch in order to compute the likelihood $\mathcal{L}_{1,n}, \mathcal{L}_{2,n}, \dots, \mathcal{L}_{M,n}$ of each Gaussian Process GP_1, GP_2, \dots, GP_M . We also use the features from the planer associated with the n^{th} board in order to compute a prediction $\hat{y}_{1,n}^*, \hat{y}_{2,n}^*, \dots, \hat{y}_{M,n}^*$ from each Gaussian Process. We then compute weights $w_{1,n}, w_{2,n}, \dots, w_{M,n}$ for each prediction according to the likelihood of the associated Gaussian Process in order to compute the final prediction of the thickness \hat{y}_n^* .

the process and performance constraints, using a single GP would likely prove inefficient in practice (the poor predictive performance of that approach is also highlighted by our results). However, using an ensemble of GPs, trained on partitions of a larger dataset (i.e., the batches), allows for a non-negligible reduction in computational complexity and gains in scalability versus using a single GP trained on the whole dataset (Tresp, 2000). Consequently, we train a different GP for each batch of the training dataset.

For a new batch, we feed the features of the current planer/board into each model and each one produced a prediction. After seeing the actual thickness of the board, the weight associated with each GP is dynamically adjusted to represent how well the GPs matches the batch. Similar weighting schemes have recently produced interesting results in many control applications, e.g., (Nguyen-Tuong et al., 2009; Schneider and Ertel, 2010; Gramacy and Apley, 2015).

In our case, we dynamically adjust the weights of the ensemble each time a new board is seen. Previous work has been done on dynamic weights, especially to model non-stationary data (Kolter and Maloof, 2007; Yin et al., 2015; Li et al., 2020). In our case, the adjustment is made probabilistically by weighting each GP proportionally to its likelihood (how well it fits the boards previously seen in the current batch). Thus it probabilistically “matches” the new batch to a convex combination of our models

trained on past batches. We named this weighting scheme **Automatic State Matching (ASM)**. The procedure is illustrated in Figure 3 and Figure 4.

More formally, as our prediction for y_n^* , the thickness of the n^{th} board, we use the mean of the predictive distribution of the ensemble, which is obtained by using the weighted individual model outputs induced by x_n^* , the vector representing the current features of the planer. The prediction produced by our ensemble follows a univariate Gaussian distribution with mean:

$$\hat{y}_n^* = \sum_{i=1}^M w_{n,m_i} K_i(x_n^*, X_i) [K_i(X_i, X_i) + \sigma_i^2 I]^{-1} Y_i, \quad (1)$$

where X_i , Y_i and K_i are respectively a vector of the features used as input during the training of the model m_i , the actual thicknesses of the associated outgoing boards and the kernel function of the model m_i as described in (Rasmussen, 2004). The weights for each model m_i for the n^{th} board, w_{n,m_i} , are computed after each board as follows:

$$w_{n,m_i} = \frac{\mathcal{L}(H_n | m_i)}{\sum_{i'=1}^M \mathcal{L}(H_n | m_{i'})}, \quad (2)$$

where \mathcal{L} is the likelihood function (Rasmussen, 2004) and H_n is the history of the batch (the $n - 1$ pairs of input vectors and thicknesses (x_j^*, y_j^*) where $j = 1, \dots, n - 1$).

Table 1. Features used as inputs for each individual Gaussian Process, whether they are adjustable or not, and source of the feature

Variable	Adjustable?	Source
Desired Position of Knives 1	Yes	Planer
Desired Position of Knives 2	Yes	Planer
Desired Position of Bedplate	Yes	Planer
Desired Position of Lower Roller	Yes	Planer
Desired Position of Upper Roller	Yes	Planer
Desired Position of Right Roller	Yes	Planer
Desired Position of Left Roller	Yes	Planer
Current OFFSET value	Yes	Planer
Speed of Knives Rotation	Yes	Planer
Desired Pressure of Upper Roller	Yes	Planer
Speed of Upper Roller	No	Planer
Line Speed	No	Planer
Real Position of Upper Roller	No	Planer
Real Position of Lower Roller	No	Planer
Temperature of Lower Roller	No	Planer
Temperature of Incoming Board	No	Sensor
Temperature of Planer	No	Sensor
Thickness of Incoming Board	No	Sensor

4. EXPERIMENTS

The data for 118,792 boards (83 batches) was gathered during a 13-day operation period. We split our dataset into a training set consisting of the data from the first 10 days and a testing set containing the data for the last 3 days. As a result, we have the data for 97,512 boards in our training set and 21,280 boards in our testing set. Our training set contains 73 training batches and our test set contains 10 batches.

Many of the features are redundant. Experts identified 10 adjustable settings that are key in optimizing thickness. We also identified 8 key informative features through data exploration and by discussing with the manufacturer. These 18 (10 + 8) features are the input of our model and the most reliable sensor for thickness will be our target variable. Table 1 summarizes the features.

In order to obtain our ensemble, a Gaussian Process is fitted on each training batch. It uses an **Automatic Relevance Determination** (ARD) Matérn kernel in order to identify which features are relevant to the batch being trained (MacKay et al., 1998). ARD kernels have been successfully used in contexts such as predicting lithium-ion batteries aging (Liu et al., 2019), soft-sensor modeling (Huazhong, 2007), and optimization of time-dependent fermentation control strategies (Coleman and Block, 2006).

For each batch in the testing set, we evaluate the mean absolute error (MAE) of our predictions w.r.t. the raw output thickness data. This metric is targeted in practice by domain experts hence our choice.

As recommended by experts from the industry, we will use as a baseline the practice of simply supposing that a board will have the same thickness as the previous board (hereafter “Last Board”). Based on their recommendation, we will also evaluate the performance of a moving average model with a window of size 5.

As a comparison, we report the results obtained by using an equal weight for each model (GPE) and the results obtained by using the sole model for which the ASM weighting scheme assigns the largest weight (Best GP). Finally, we tried many models from the machine learning literature trained on the whole training dataset, notably many variations of linear regression, support vector machines, decision trees and neural networks. Among these “classical” models, we present only the results of XGBoost since it provided the best results.

5. RESULTS

Table 2 shows the MAE for different approaches applied on the 21,280 boards of the 10 batches from the test set. Our baseline (**Last Board**) is in light gray and our approach (**ASM-GPE**) is in dark gray. The last line of the table is the average performance for all the batches of the test, weighted by board counts.

First, the moving average model (MA-5) led to an increase of the error of 14.44% in comparison to the baseline (**Last Board**). This confirms the intuition of the experts that predicting the thickness of the previous board is a better baseline.

Training a single model (**XGBoost**) or using an equal weight in an ensemble of Gaussian Processes (**ASM-GPE**) also produces an increase of the MAE (respectively 20.27% and 59.28%). Due to the high variability observed between the batches, this was expected.

Selecting the single “best” model from the GPs ensemble (**Best GP**) allows a reduction in the MAE of 27.25% on average. However, for Batch 4 the MAE is still lower for the **Last Board** model.

Finally, our approach (**ASM-GPE**) produced the best results with the lowest MAE for every batch and an average reduction of the MAE of 39.63%.

6. CONCLUSION

ASM-GPE allows for some unique advantages in our context, namely we explicitly know which past batches collectively best matches the current one. In practice, this valuable information can be used in various ways, e.g., raising flags when the weight associated with a GP trained on a past batch known to have preceded mechanical issues exceeds a certain threshold. In addition, our approach produced the best results with the lowest MAE for every batch, an average reduction of the MAE of 39.63% w.r.t. the current standard using the **Last Board** thickness as a predictor. As a result, the approach is currently being implemented in an actual mill to suggest “OFFSET” adjustments to the operators. After this first implementation, the full automated control loop will be implemented.

Finally, although the approach is currently being used to adjust a single variable, i.e., the offset, it is designed to be able to assist in the optimization of all settings by extracting information as to which adjustable settings have the greatest impact in the current state. We can extract this information from the most likely GPs’ ARD Kernel and then use the model itself alongside classical

Table 2. MAE for the **Last Board** baseline for 10 test batches and relative MAE diminution (in %) for the **MA-5**, **XGBoost**, **GPE**, **Best GP** and **ASM-GPE** models

Batch	Boards (#)	MAE		Change in MAE (%)				
		Last Board	MA-5	XGBoost	GPE	Best GP	ASM-GPE	
1	1852	0.00887	↓ 2.78 %	↑ 51.74 %	↑ 70.68 %	↓ 35.62 %	↓ 38.16 %	
2	1257	0.01728	↑ 0.57 %	↓ 2.55 %	↑ 56.49 %	↓ 67.76 %	↓ 71.42 %	
3	2625	0.01618	↑ 15.28 %	↑ 24.88 %	↑ 58.65 %	↓ 29.44 %	↓ 42.68 %	
4	1834	0.01147	↑ 24.34 %	↑ 45.74 %	↑ 67.13 %	↑ 3.45 %	↓ 18.36 %	
5	787	0.00939	↓ 3.98 %	↑ 47.36 %	↑ 73.23 %	↓ 48.81 %	↓ 58.34 %	
6	504	0.02169	↓ 1.16 %	↑ 6.46 %	↑ 44.69 %	↓ 78.51 %	↓ 83.96 %	
7	1299	0.01909	↓ 1.70 %	↓ 5.70 %	↑ 48.81 %	↓ 72.44 %	↓ 81.11 %	
8	3132	0.01537	↑ 22.95 %	↑ 27.22 %	↑ 57.68 %	↓ 13.51 %	↓ 19.70 %	
9	6949	0.01624	↑ 19.64 %	↑ 11.59 %	↑ 58.97 %	↓ 19.23 %	↓ 32.78 %	
10	1041	0.01748	↓ 3.80 %	↓ 43.04 %	↑ 50.79 %	↓ 47.01 %	↓ 74.27 %	
Total	21280	0.01522	↑ 14.44 %	↑ 20.27 %	↑ 59.28 %	↓ 27.25 %	↓ 39.63 %	

optimization techniques in order to find the values for the settings that would result in the highest quality boards. Further research will focus on demonstrating the potential of ASM-GPE in this regard.

ACKNOWLEDGEMENTS

The authors would like to thank the FORAC Research Consortium and its partners. Our gratitude goes as well to the Natural Sciences and Engineering Research Council of Canada (NSERC) who provided funding for this research.

REFERENCES

- Barto, A.G. (1994). Reinforcement learning control. *Current opinion in neurobiology*, 4(6), 888–893.
- Camacho, E.F. and Alba, C.B. (2013). *Model predictive control*. Springer science & business media.
- Coleman, M.C. and Block, D.E. (2006). Retrospective optimization of time-dependent fermentation control strategies using time-independent historical data. *Biotechnology and bioengineering*, 95(3), 412–423.
- Gramacy, R.B. and Apley, D.W. (2015). Local gaussian process approximation for large computer experiments. *Journal of Computational and Graphical Statistics*, 24(2), 561–578.
- Huazhong, W. (2007). Gaussian process and its application to soft-sensor modeling [j]. *Journal of Chemical Industry and Engineering (China)*, 11.
- Johnson, M.A. and Moradi, M.H. (2005). *PID control*. Springer.
- Kocijan, J. (2016). *Modelling and control of dynamic systems using Gaussian process models*. Springer.
- Kolter, J.Z. and Maloof, M.A. (2007). Dynamic weighted majority: An ensemble method for drifting concepts. *The Journal of Machine Learning Research*, 8, 2755–2790.
- Li, H., Wang, J., and Yang, H. (2020). A novel dynamic ensemble air quality index forecasting system. *Atmospheric Pollution Research*, 11(8), 1258–1270.
- Lillicrap, T.P., Hunt, J.J., Pritzel, A., Heess, N., Erez, T., Tassa, Y., Silver, D., and Wierstra, D. (2015). Continuous control with deep reinforcement learning. *arXiv preprint arXiv:1509.02971*.
- Liu, K., Li, Y., Hu, X., Lucu, M., and Widanage, W.D. (2019). Gaussian process regression with automatic relevance determination kernel for calendar aging prediction of lithium-ion batteries. *IEEE Transactions on Industrial Informatics*, 16(6), 3767–3777.
- MacKay, D.J. et al. (1998). Introduction to gaussian processes. *NATO ASI series F computer and systems sciences*, 168, 133–166.
- Nguyen-Tuong, D., Seeger, M., and Peters, J. (2009). Model learning with local gaussian process regression. *Advanced Robotics*, 23(15), 2015–2034.
- Rasmussen, C. (2004). *Gaussian Processes in Machine Learning*. Springer.
- Schneider, M. and Ertel, W. (2010). Robot learning by demonstration with local gaussian process regression. In *2010 IEEE/RSJ International Conference on Intelligent Robots and Systems*, 255–260. IEEE.
- Tresp, V. (2000). A bayesian committee machine. *Neural computation*, 12(11), 2719–2741.
- Usuga Cadavid, J.P., Lamouri, S., Grabot, B., Pellerin, R., and Fortin, A. (2020). Machine learning applied in production planning and control: a state-of-the-art in the era of industry 4.0. *Journal of Intelligent Manufacturing*, 31(6), 1531–1558.
- Yin, X.C., Huang, K., and Hao, H.W. (2015). De2: Dynamic ensemble of ensembles for learning nonstationary data. *Neurocomputing*, 165, 14–22.