Real-Time Parameter Estimation of Central Air Handling Unit: Algebraic and Recursive Least Squares Techniques

Xing-Yi Li*†, Danielle Nyakam Nya†, Franco Falconi *, Tarek Raissi †
† Conservatoire National des Arts et Métiers (CNAM), EA4629, CEDRIC, Paris, France
* Scheinder Elecrtic, Rueil Malmaison, France
Emails: (xingyi.li, danielle.nyakam-nya)@lecnam.net, franco.falconi@se.com, tarek.raissi@cnam.fr

***Abstract*—This paper presents a comparative study of two parameter identification methods: an algebraic approach based on the direct manipulation of the system's differential equations and the Recursive Least Squares (RLS) technique. The objective is to evaluate the performance of both methods in terms of estimation accuracy, noise robustness, and computational complexity. The methods are applied to a linear time-invariant system in the programmable logic controller (PLC) environment. Results show that the algebraic approach, although non-recursive, offers fast and robust estimates under moderate noise conditions, while the RLS excels in real-time adaptability in dynamic scenarios. The findings are discussed concerning real-time implementation constraints and potential applications, particularly in industrial systems: Central Air Handling Unit (AHU).**

***Keywords*–AHU, industrial systems, Algebraic estimation, module theory, differential algebra, operational calculus, RLS**

I. Introduction

In the field of process control, accurate systems identification plays a crucial role in modeling, simulation, and controller design. Two main paradigms are commonly adopted: the continuous-time approach, based on differential equations, and the discrete-time approach, which relies on sampled data and difference equations. The choice between the two depends on the system's nature, implementation requirements, and available data. In this study, we focus on the identification of dynamic models for AHU systems, comparing the RLS method and an algebraic identification technique. The paper aims to provide insight into the strengths and limitations of each method, particularly in the context of real-time applications.

A. Typical Structure of an AHU

An AHU is made up of several devices, as shown in Figure 1. It includes cooling and/or heating coils, air dampers, supply/exhaust air fans, and several measurement tools.

In the diagram, indoor air is extracted by an exhaust fan through the blue tube. It passes through a filter that ensures the removal of solid particles and unwanted impurities from entering the AHU. The PT and PE sensors in the filter detect the pressure difference before and after the filter, thus ensuring that air passes through the filter without issues and maintaining air quality. Downstream, air flows through the extraction fan,

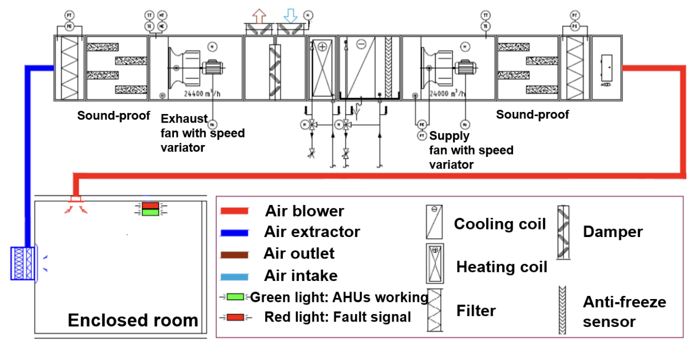

Fig. 1: Diagram of AHU structure

which is responsible for air extraction and is equipped with a variable-speed controller to control the flow rate. In the AHU extraction mode, the damper between the fresh air box and the extraction box is closed, allowing air to exit through the extraction duct. In normal mode, AHUs introduce fresh air by opening the fresh air damper, thus promoting oxygen renewal in enclosed spaces and eliminating odours to ensure air quality. Depending on operational requirements and control objectives, it is possible to choose to use all fresh air or to mix fresh air with return air extracted by the extraction fan by opening and closing the dampers between the two boxes. After the fresh air is recovered, it enters the heating coil and the cooling coil. The heating coil and cooling coil contain hot or cold water in the coil for heat exchange with the air. The hot water and cold water valves control the flow of hot and cold water, thereby regulating the amount of heat and cold exchanged with the air. The supply fan propels the air at variable speeds to adjust the flow rate, then returns it to the enclosed space through the filter in the red tube. Additionally, the antifreeze thermostat (antifreeze sensor) protects the cooling coil by detecting ice formation in the cold water pipe. With the speed-variator included in the supply/exhaust fan, air-handling units can be separated into variable-air-volume (VAV) air-handling units or constant-air-volume (CAV) air-handling units. The supply air temperature is the most crucial factor for controlling the AHU [1] .

B. Analysis of dynamic control modeling

In industrial projects, engineers control supply air temperature changes by controlling the amount of hot and cold water entering the coils and the amount of air supplied (the input $u(t)$). The majority of air conditioning systems suffer from information disturbances and deficits and have inherent dynamic/static nonlinearities and time-varying environments. It has therefore been accepted that the air-conditioning system is among the most challenging plants in the field of process control [2].

Advanced control techniques have been proposed to enhance the performance of air conditioning systems in terms of robust stability, energy efficiency, operational reliability, and other attributes. These techniques include self-tuning control [3], fuzzy control [4], neural network control [5], and predictive control [6].

To control the supply air temperature (which can be denoted by $y(t)$). We then need to find a model between the control of the hot and cold water valves $u(t)$ and the temperature of the supply air $y(t)$.

The air-conditioning industry has been slow to adopt replacements for PID. Generally speaking, the building industry is hesitant to embrace something that could need to be handled like a black box. PID is nevertheless commonly utilised today [2]. So we need to find the model with the simplest terms to facilitate the control processes in the industries. At the same time, to facilitate timely observation of system changes, the online identification should be used in the programme.

C. Comparison of continuous-time and discrete-time modeling approaches for AHU systems

Two main approaches can be considered for process control and, consequently, for systems identification [7]. The first, and most intuitive, is the *continuous-time approach*, which relies on a continuous description of the system's temporal behavior using differential equations [8], [9]. This framework is often based on physical laws, such as energy and mass balances, and is widely used in the modeling of thermal and fluid systems, including the Heating, Ventilation and Air Conditioning (HVAC) applications. The second is the *discrete-time approach*, which explicitly considers the sampling process inherent to digital measurement and control systems. It describes the system evolution through difference equations and is well-suited for digital implementation and real-time control [10], [11]. This approach is especially relevant when working with data-driven models.

D. organisation

The remainder of this paper is organized as follows. Section II briefly recalls the real-time estimation via algebraic techniques setting. Section III presents the estimation of parameters using RLS. Section IV completed a comparison analysis based on two online parameter identification techniques by examining systems that basic models can characterise. Finally, Section V gives some concluding remarks and perspectives for future research.

II. Short Overview of Algebraic Techniques

This approach, which exhibits good robustness properties with respect to a large variety of additive perturbations, is based on module theory, differential algebra and operational calculus (see [12], [13]). Its algebraic nature permits the derivation of exact non-asymptotic formulae for obtaining the unknown quantities in real time. There is no need to know the statistical properties of the corrupting noises and the initial values, [14], [15], [16], [17].

A. Plan modeling AHU

As is common in industrial practice, the HVAC system is modelled as a first-order damped system without time delay:

$$\tau \dot{y}(t) + y(t) = Ku(t) + \zeta_0 \tag{1}$$

The parameters K is the static gain, the τ is the time constant and the ζ_0 is the disturbance that may be generated by the measurement device or signal. In the real system, its form depends on the equipment, while in the simulated environment, it is represented as Gaussian white noise.

B. Parameter identification using algebraic approach

Now we are interested in estimating the parameters K and τ of model of AHU system defined by the differential equation (1). Based on the following steps:

- Step 1 : Operational domain

Applying Laplace transforms [18] to (1) leads to (2):

$$\tau s y(s) - Ku(s) + y(s) = \tau y(0) + \frac{\zeta_0}{s} \tag{2}$$

- Step 2 : Algebraic manipulation

For the algebraic manipulation, to eliminate the initial conditions and the perturbation ζ_0 , both sides of equation (2) by s and differentiate twice with respect to s using Leibniz's formula , we get our first equation:

$$\tau s^2 y(s) - Ksu(s) + sy(s) = \tau s y(0) + \zeta_0 \tag{3}$$

Taking the derivative with respect to $\frac{d}{ds}$ gives:

$$\left(2sy(s) + s^2\frac{dy}{ds}\right)\tau - \left(u(s) + s\frac{du}{ds}\right)K + y(s) + s\frac{dy}{ds} = \tau y(0) \tag{4}$$

By differentiating equation (4) again with respect to $\frac{d}{ds}$, we eliminate the initial condition $y(0)$.

This leads to the first equation:

$$\left(s^2\frac{d^2y}{ds^2} + 4s\frac{dy}{ds} + 2y(s)\right)\tau - \left(2\frac{du}{ds} + s\frac{d^2u}{ds^2}\right)K = -2\frac{dy}{ds} - s\frac{d^2y}{ds^2} \tag{5}$$

- Step 3: Redundancy equation

Since there are two parameters to estimate, a second equation is needed to obtain a system with two unknowns. Differentiating equation (5) with respect to s yields the second equation. This results in a system of two unknowns:

$$\underbrace{\begin{pmatrix} s^2\frac{d^2y}{ds^2}+4s\frac{dy}{ds}+2y(s) & -2\frac{du}{ds}-s\frac{d^2u}{ds^2} \\ s^3\frac{d^3y}{ds^3}+6s^2\frac{d^2y}{ds^2}+6s\frac{dy}{ds} & -3\frac{d^2u}{ds^2}-s\frac{d^3u}{ds^3} \end{pmatrix}}_{\aleph(s)} \underbrace{\begin{pmatrix} \tau \\ K \end{pmatrix}}_{\Theta} = \underbrace{\begin{pmatrix} -2\frac{dy}{ds}-s\frac{d^2y}{ds^2} \\ -3\frac{d^2y}{ds^2}-s\frac{d^3y}{ds^3} \end{pmatrix}}_{\beta(s)} \quad (6)$$

To avoid using derivatives with respect to time (positive powers of s), which are sensitive to noise, we multiply both sides of equation (6) by $s^{-\nu}, \nu \in \mathbb{R}$ (with $\nu = 3$):

For numerical implementation reasons, a final step involves transforming back into the time domain using standard inverse Laplace transforms, ensuring that only integrations of the measured signals are involved:

- Step 4: Return to the time domain
  - Multiple integrals: $s^{-n} \rightarrow \int^{(n)}$ and $s^{-1} \rightarrow \int_0^t$
  - Algebraic derivative: $\frac{d^n}{ds^n} \rightarrow (-t)^n$ and $\frac{d}{ds} \rightarrow -t$
  - Example: $\int^{(3)} u(t) = \int_0^T \int_0^{t_2} \int_0^{t_1} u(\tau)dt_1\, dt_2\, d\tau$

The final result is:

$$\Theta = \aleph^{-1}(t).\beta(t) \quad (7)$$

Where,

$$\aleph(t) = \begin{pmatrix} \int^{(1)} t^2y - 4\int^{(2)} ty + 2\int^{(3)} y & h_{12} \\ -\int^{(1)} t^3y + 6\int^{(2)} t^2y - 6\int^{(3)} ty & h_{22} \end{pmatrix}$$

$$\Rightarrow \begin{cases} h_{12} = 2\int^{(3)} tu - \int^{(2)} t^2u \\ h_{22} = -3\int^{(3)} t^2u + \int^{(2)} t^3u \end{cases}$$

$$\beta(t) = \begin{pmatrix} 2\int^{(3)} ty - \int^{(2)} t^2y \\ -3\int^{(3)} t^2y + \int^{(2)} t^3y \end{pmatrix}$$

III. Recursive Least Squares

Least squares is a commonly used linear regression algorithm. It is used widely in systems identification and parameter estimation. The least square estimation still can provide an accurate solution when other identification methods lose efficacy [19].

Let X_k be the regressor constructed with the data from inputs and outputs up to time k i.e.:

$$X_k = \begin{bmatrix} x_{11} & x_{12} & \cdots \\ x_{21} & x_{22} & \cdots \\ x_{31} & x_{32} & \cdots \\ \cdots & \cdots & x_{kn} \end{bmatrix} = \begin{bmatrix} \phi_1 \\ \phi_2 \\ \cdots \\ \phi_k \end{bmatrix} \quad (8)$$

The estimated Θ obtained at time k is then given by

$$\Theta_k = (X_k^T X_k)^{-1} X_k^T Y_k \quad (9)$$

Let us denote $P_k^{-1} = X_k^T X_k$ with $P(0) = \delta I \in C^{M\times M}$, where δ is a positive value chosen for the initialisation of the algorithm, and let $\Theta_0 = 0$. Then:

$$X_k^T X_k = \begin{bmatrix}\phi_1 & \cdots & \phi_k\end{bmatrix} \begin{bmatrix} \phi_1^T \\ \vdots \\ \phi_k^T \end{bmatrix} = \sum_{i=1}^{k-1} \phi_i\phi_i^T + \phi_k\phi_k^T \quad (10)$$

$$P_k^{-1} = X_k^T X_k = P_{k-1}^{-1} + \phi_k\phi_k^T \quad (11)$$

$$X_k^T Y_k = X_{k-1}^T Y_{k-1} + \phi_k y_k \quad (12)$$

According to the ordinary least squares form:

$$\Theta_{k-1} = (X_{k-1}^T X_{k-1})^{-1} X_{k-1}^T Y_{k-1} = P_{k-1} X_{k-1}^T Y_{k-1}$$
$$P_{k-1}^{-1}\Theta_{k-1} = X_{k-1}^T Y_{k-1} \quad (13)$$

With equations (11), (12), (13), Θ_k rewrited as:

$$\Theta_k = \Theta_{k-1} + K_k\varepsilon_k \quad (14)$$

with

$$K_k = P_k\phi_k = P_{k-1}\phi_k[1 + \phi_k^T P_{k-1}\phi_k]^{-1} \quad (15)$$
$$\varepsilon_k = y_k - \phi_k^T\Theta_{k-1} \quad (16)$$
$$P_k = (P_{k-1}^{-1} + \phi_k\phi_k^T)^{-1} = [I - K_k\phi_k^T]P_{k-1} \quad (17)$$

I as the identity matrix.

Using the Woodbury matrix identity [20], in (17), we decompose the inverse of the sum of two matrices. Then, we obtain the recursive form of equations (14), (15), and (16). To ensure the recursive process converges and emphasise the impact of new data on results, a forgetting factor λ ($0 < \lambda \leq 1$) is often added. We then obtain:

$$K_k = \frac{\lambda^{-1}P_{k-1}\phi_k}{1 + \lambda^{-1}\phi_k^T P_{k-1}\phi_k} \quad (18)$$

$$P_k = \lambda^{-1}P_{k-1} - \lambda^{-1}\frac{P_{k-1}\phi_k\phi_k^T P_{k-1}}{1 + \phi_k^T P_{k-1}\phi_k} \quad (19)$$

The k^{th} output of the system is determined by

$$y(k) = ay(k-1) + bu(k) \quad (20)$$

If we regress the system using a first-order system and the RLS method, the Y_k is the k^{th} measurement and $y(k-1)$ and $u(k)$ are known quantities that construct X as x_{k1}, x_{k2}, they can also be written as $\phi_k = [y(k-1)\ u(k)]^T$. The unknown parameter vector $\Theta = [a\ b]^T$, the solution of which is updated with new results at each sampling time using the RLS formula (14), (17), (18) and (19).

Once we get the result of the parameter vector $\Theta = [a\ b]^T$, with the sample time T_s, We relate the discrete transfer function (20) to the first-order continuous transfer function using Backward Euler method.Then the correspondence between the parameters [21] :

$$a = \frac{\tau}{\tau + T_s};\ b = \frac{K\cdot T_s}{\tau + Ts} \rightarrow K = \frac{b}{1-a};\ \tau = \frac{a\cdot T_s}{a-1}$$

IV. Discussions

A simulated first-order AHU process was constructed in programmable logic controller programming software, with a transfer function $\frac{5.0}{10s+1}$. Input a step signal $u(t)$, with a sampling period of $T_s = 0.5$ second and a length of 3731 effective values as shown in Fig 2. The output $y(t)$ is shown in Fig 3.

This simulation was implemented using EcoStruxure Control Expert, the Schneider programmable controller programming software commonly used in industrial process control projects, version 15.2. It was done using the Schneider programmable controller M340-BMX-P34-2020. Calculating the online parameters of the system uses algebraic and RLS techniques. Different possible input signals were stimulated to observe the differences in the results of the two techniques. In the algebraic technique, higher-order integral calculations are used to minimise unwanted numerical fluctuations in the PLC controller due to numerical divergence, it is also a solution for algebraic methods to address noisy systems. It can help the system achieve stability and more accurate parameter output values [22]. This simulation superimposes five more integrals on the elements of Eq. (7).

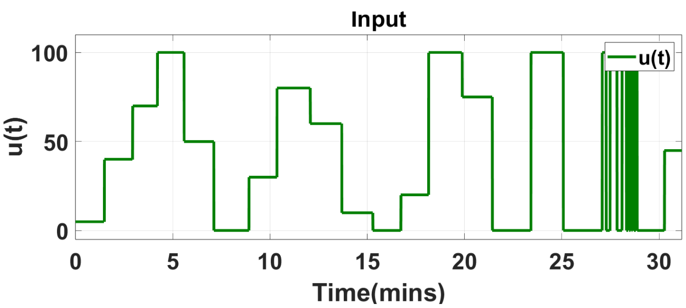

Fig. 2: Input $u(t)$

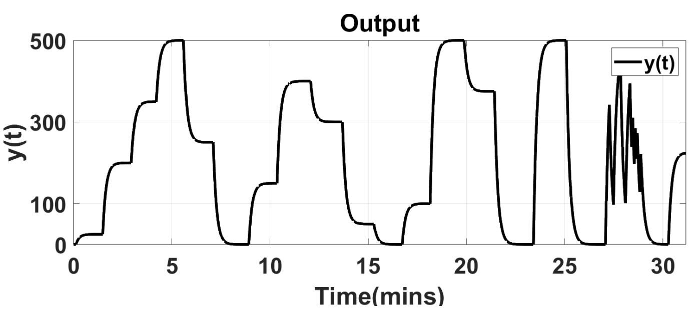

Fig. 3: Output $y(t)$

Simultaneous online identification of system parameters is performed using the inputs $u(t)$ and outputs $y(t)$ of the system, combined with the two parameter identification methods of Part II and Part III.

When using RLS to calculate parameters, based on the explanation in the Part III, to ensure the system has relatively online update capability, it λ should not be chosen very close to 1 (the experiment uses 0.9). Two situations require special attention. When the output reached steady state, τ could not maintain the relatively correct value calculated during the transit regime. On the other hand, for the parameter K, if the system input is 0, RLS also finds it difficult to obtain a relatively accurate K. A solution is to add a program segment in the PLC program to check if the system input $u(t)$ is 0 or the output has reached a stable state. Once meet these conditions, pause the update of the RLS parameter calculation output; and as soon as a new step command is received, keep updating the computing values.

The above phenomenon can be observed in Fig 4 and Fig 5 obtained from similar experiments. This issue is more pronounced in the case of system superimposed noise Fig 8.

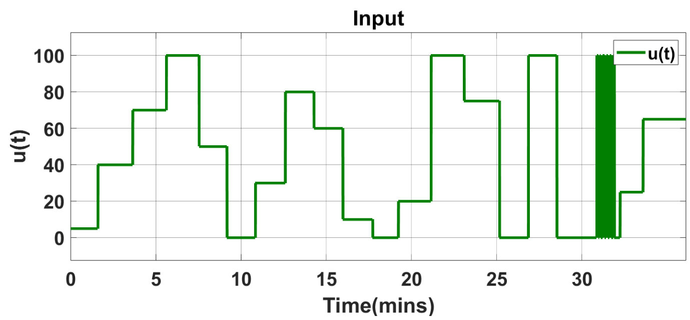

Fig. 4: Similar input $u(t)$ as Fig2

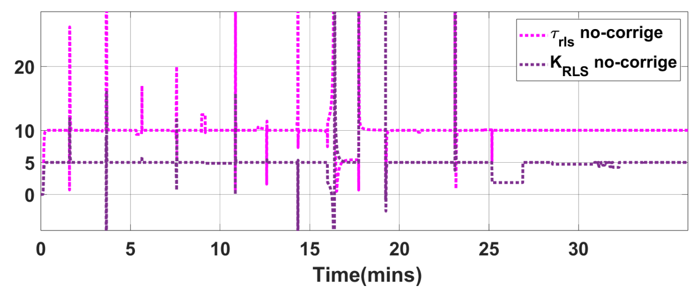

Fig. 5: Using RLS calculate $K$and τ

The corriged variations of the gain K and the time-constant parameter τ are shown in Figures 6 and 7.

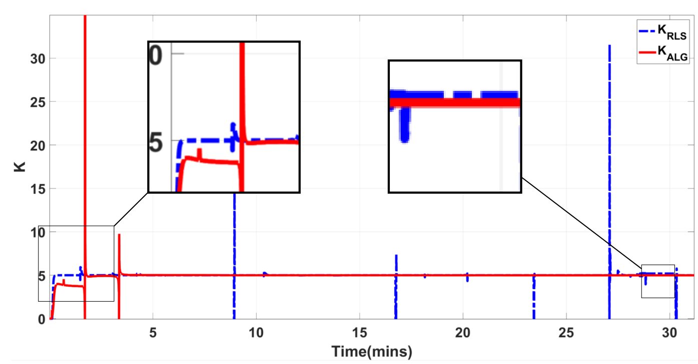

Fig. 6: Calculated K after correction

In figures 6 and 7, the solid red lines represent the parameter variations calculated using the algebraic technique. As it uses the integral calculation from time 0 to the current time t to obtain the parameters.

When the recording duration is short, the algebraic approach

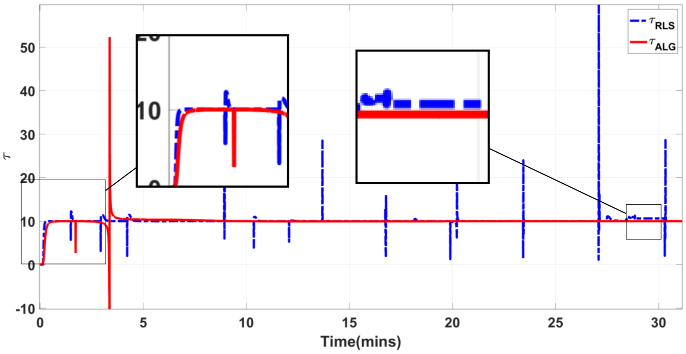

Fig. 7: Calculated τ after correction

is more likely to compute parameter values that differ slightly from the real values and is more susceptible to numerical fluctuations. But when the program's execution time grows, It starts giving very consistent results for the parameter computations that are close to the right values. Even in the latter part of the simulation, the inputs and outputs of the system are highly variable and show irregular, extreme value jumps. With minimal impact, the algebraic approach can continue to produce sufficiently precise values.

Meanwhile, the blue dotted broken lines all represent the change in value of the same input calculated by RLS. RLS can quickly find the correct parameters of the system during the transit regime of the simulation step. At this stage, RLS is faster and more accurate than algebraic methods. But RLS is relatively sensitive to the input and output variations of the system. At the beginning of each step, the values calculated by RLS fluctuate for a short period, resulting in outliers far from the correct values. In the later stages of the simulation, the output of RLS begins to produce a persistent error as the input and output change rapidly. After the system reverts back to zero input, RLS itself is unable to quickly correct the parameter calculation errors. However, if a non-zero input is reintroduced to the system at this point, RLS can again find the correct parameter values for the system.

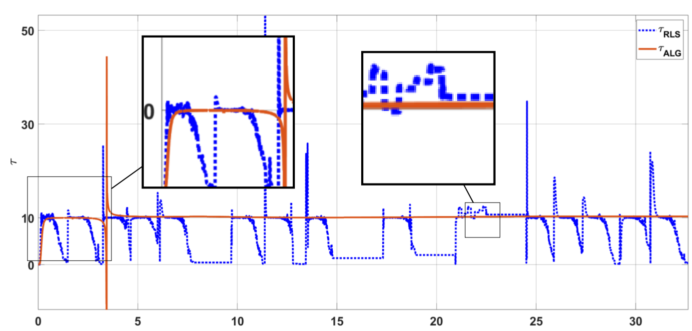

Fig. 8: Calculated τ in noisy environments

Considering the complexity of the industrial site environment, introduce random values in the [-3 3] as output section disturbances. Do the experiment again, giving the system various combinations of step signals $u(t)$ as inputs. Get the result $y_{Noisy}(t)$ as system output.

In a noisy environment, the algebraic method has better parameter output stability, but requires a period of experimentation to achieve results sufficiently close to the set value. RLS produces more parameter output value jitter in a noisy environment, and for the parameter τ, with the same criteria in a noiseless situation, it is difficult to stabilize at the correct value in the steady state. If we also add that RLS stops updating when the input $u(t)$ is 0, it becomes easy for the calculated τ to output incorrect values for a long period. This greatly affects our system evaluation. In an industrial environment, we cannot accurately know the amplitude of the noise. More experiments and observations are needed to define the criteria for judging that the output has entered a stable state. If we encounter a system with very slow output changes and some noise overlay, we should consider a more complex judgment.

After optimizing the correction conditions, for similar inputs $u(t)$, relatively stable parameter output results shown in Fig 9 and Fig 10 can be obtained under the two identification methods.

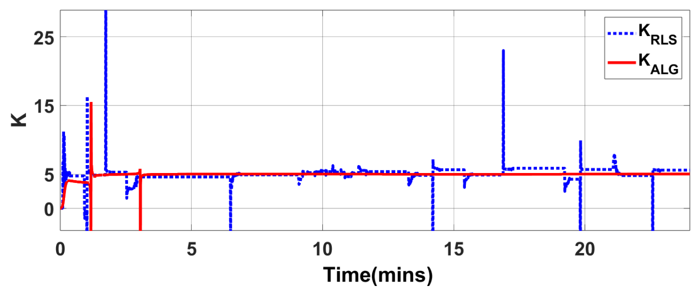

Fig. 9: Calculate K with noise after correction

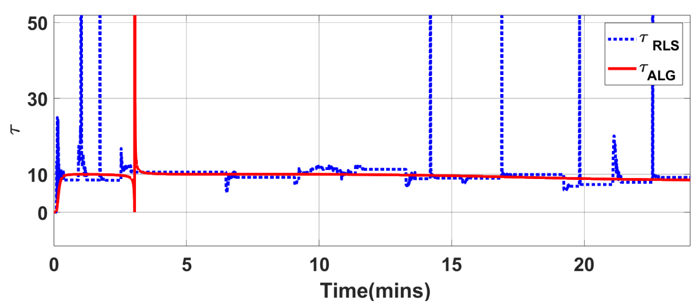

Fig. 10: Calculate τ with noise after correction

We use the Mean Square Error (MSE), Mean Absolute Error (MAE), Root Mean Squared Error (RMSE) and Mean Absolute Percentage Error (MAPE) as the four evaluation metrics to determine the difference between the calculated values and the set values. The data used for the table calculations are the output-stabilised optimised data from Fig 6, 7, 9, and 10.

As can be seen from the graphs and table I, the algebraic method provides significantly more stable results than RLS after the system has run for a certain period in a programmable logic controller environment. It has a strong ability to identify stable systems whose parameters do not vary with time. The numerical output is relatively stable regardless of the presence

TABLE I: Comparative study of estimation techniques

Critères	Noise	Approche algèbrique	RLS
MSE	$y(t)$	K=42.0075 τ=1.8964	K=0.6848 τ=1.0379e+08
	$y_{Noisy}(t)$	K=0.8810 τ=103.1912	K=4.8183 τ=2.2682e+08
MAE	$y(t)$	K=0.1981 τ=0.1930	K=0.0764 τ=1.0088e+04
	$y_{Noisy}(t)$	K=0.1181 τ=0.7373	K=0.5792 τ=1.0059e+03
RMSE	$y(t)$	K=6.4813 τ=1.3771	K=0.8275 τ=1.0188e+04
	$y_{Noisy}(t)$	K=0.9386 τ=10.1583	K=2.1951 τ=1.5061e+04
MAPE	$y(t)$	K=3.9610% τ=1.9305%	K=1.5282% τ=1.0088e+05%
	$y_{Noisy}(t)$	K=2.3612% τ=7.3731%	K=11.5840% τ=1.0059e+05%

of noise. Even in the presence of strong noise, the effect of noise on the identified system can be counteracted by increasing the number of integrals. The errors and fluctuations generated by the algebraic method may give poor results in outlier-sensitive evaluation metrics such as MSE, RMSE, etc., when the system is just starting to run and there is not yet enough data.

At the same time, the output results of RLS are quite sensitive to changes in system inputs and outputs and fluctuate whenever the value of the system input u(t) changes. In the absence of noise, RLS can rediscover the correct system parameter values; in noisy environment, engineer needs to carefully confirm the identification interval. These fluctuations, when carried over to the computation of the evaluation criterion in Table (I), lead to obtain a relatively larger value.

In terms of industrial implementation, algebraic methods require high computational power of the hardware and strong numerical capabilities of the software because they require continuous calculation of integral values. If higher-order integral calculations are introduced, engineers need to pay close attention to numerical variations during programming to avoid errors such as overruns.

The benefits and drawbacks of each method are presented in the table II.

TABLE II: Summary of application performance

	Algebraic approach	**RLS**
Preferred scenario	prolonged operation	transient start-up
Advantage	stable parameter output	quick response
Weakness	high demands on the calculation range of the device	input/output sensitive
Noise solutions	higher-order integrals	limit the identification range

The same simulation can also be implemented using the Schneider M580 programmable logic controller.

V. Conclusion

This paper details two approaches to system identification in the field of process control: a continuous-time approach using differential equations and a discrete-time approach that relies on sampled data. The choice between these two approaches depends on the nature of the system, implementation requirements, and available data. Two techniques are compared in the context of identifying dynamic models of HVAC systems. The advantages and limitations of each method are explored in depth, especially for real-time applications with and without noise added. Based on the sensitivity and stability of continuous-time methods and discrete-time methods, investigate identifying variable-parameter systems with easy industrial implementation.

References

[1] G. Huang and P. Zhou, "Generalized eigenvalue minimization used in robust temperature control design for air-handling units," in *Proceedings of the 31st Chinese Control Conference*, 2012, pp. 4634–4639.

[2] T. I. Salsbury, "A survey of control technologies in the building automation industry," *IFAC Proceedings Volumes*, vol. 38, no. 1, pp. 90–100, 2005, 16th IFAC World Congress.

[3] Q. Bi, W.-J. Cai, Q.-G. Wang, C.-C. Hang, Eng-Lock Lee, Y. Sun, K.-D. Liu, Y. Zhang, and B. Zou, "Advanced controller auto-tuning and its application in hvac systems," *Control Engineering Practice*, vol. 8, no. 6, pp. 633–644, 2000.

[4] S. Hussain, H. A. Gabbar, D. Bondarenko, F. Musharavati, and S. Pokharel, "Comfort-based fuzzy control optimization for energy conservation in hvac systems," *Control Engineering Practice*, vol. 32, pp. 172–182, 2014.

[5] R. Heidarykiany and C. Ababei, "Advanced day-ahead scheduling of hvac demand response control using novel strategy of q-learning, model predictive control, and input convex neural networks," *Energy and AI*, p. 100509, 2025.

[6] G. Liu, J. Gao, Z. Han, and Y. Yuan, "Hybrid model-based predictive hvac control through fast prediction of transient indoor temperature fields," *Building and Environment*, vol. 267, p. 112253, 2025.

[7] F. Falconi, "Méthodes avancées pour l'identification, la commande et le diagnostic prédictif des systèmes complexes," Ph.D. dissertation, HESAM Université, 2021.

[8] K. Åström and R. Murray, *Feedback Systems: An Introduction for Scientists and Engineers*. Princeton University Press, 2008.

[9] A. Bejan, *Convection Heat Transfer*. Wiley, 2013.

[10] L. Ljung, *System Identification: Theory for the User*. Prentice Hall, 1999.

[11] T. F. E. Dale E. Seborg, Duncan A. Mellichamp and F. J. Doyle, *Process Dynamics and Control*. Wiley, 2010.

[12] M. Fliess, C. Join, and W. Perruquetti, "Real-time estimation for switched linear systems," in *2008 47th IEEE Conference on Decision and Control*. IEEE, 2008, pp. 941–946.

[13] M. Fliess and H. Sira-Ramírez, "An algebraic framework for linear identification," *ESAIM: Control, Optimisation and Calculus of Variations*, vol. 9, pp. 151–168, 2003.

[14] M. Fliess and H. Sira-Ramirez, "Closed-loop parametric identification for continuous-time linear systems via new algebraic techniques," in *Identification of Continuous-time Models from sampled Data*. Springer, 2008, pp. 363–391.

[15] Y. Tian, T. Floquet, and W. Perruquetti, "Fast state estimation in linear time-varying systems: an algebraic approach," in *2008 47th IEEE Conference on Decision and Control*. IEEE, 2008, pp. 2539–2544.

[16] D. N. Nya and H. Abouaïssa, "A robust inventory management in dynamic supply chains using an adaptive model-free control," *Computers & Chemical Engineering*, vol. 179, p. 108434, 2023.

[17] D. N. Nya, S. Hachour, and H. Abouaïssa, "Inventory control in supply chain: a model-free approach," *IFAC-PapersOnLine*, vol. 55, no. 10, pp. 2755–2760, 2022.

[18] M. R. Spiegel, *Laplace transforms*. McGraw-Hill New York, 1965.

[19] S. Wang, "Windowed least square algorithm based pmsm parameters estimation," *Mathematical Problems in Engineering*, vol. 2013, 01 2013.

[20] D. A. Harville, *Matrix Algebra From a Statistician's Perspective*. New York, NY: Springer, 1997.

[21] K. L. F. Rodriguez, R. R. Alvarado, M. Barati, V. S. Tagle, and R. S. V. Gonzalez, *Recent trends in sustainable engineering : proceedings of the 2nd International Conference on Applied Science and Advanced Technology*, 1st ed., ser. Lecture Notes in Networks and Systems ; Volume 297. Cham, Switzerland: Springer, 2022.

[22] C. Join, "Une approche algébrique pour la pratique de l'estimation, du diagnostic, de la commande et de la finance," Habilitation à diriger des recherches, Université de Lorraine, Jun. 2012.