Linear Parametric Identification for a Servomechanism in a Mechanical Ventilator System

José Pablo Cabrera Flores
School of Engineering
Universidad Anáhuac Querétaro
josepablocf@hotmail.com

Gerardo Rojas Soltero
School of Engineering
Universidad Anáhuac Querétaro
gerard-rs1@hotmail.com

Carolina Cristerna Mondragón
School of Engineering
Universidad Anáhuac Querétaro
carolina_cmon@hotmail.com

Herubey Acosta Arellano
School of Engineering
Universidad Anáhuac Querétaro
heru1028@hotmail.com

Mario Jafet Áviles Blanco
School of Engineering
Universidad Anáhuac Querétaro
mariojafet@live.com

Josue Ojeda Villegas
School of Engineering
Universidad Anáhuac Querétaro
chin_2323@hotmail.com

Alberto Alonso Arellano
School of Engineering
Universidad Anáhuac Querétaro
albertoare_a@live.com.mx

Eloy Edmundo Rodríguez-Vázquez
CIDESI & U. Anáhuac
eloy.rodriguez@ieee.org

***Abstract*—. Covid-19 has caused a health and economic crisis worldwide. Unfortunately, the lack of medical equipment, such as ventilators, represent a significant obstacle for the medical community to treat patients suffering from this virus, therefore we decided to create an algorithm that can correctly estimate linear parameters for an auxiliary mechanical ventilation system with the purpose of helping developers to filter respiratory leaks from a signal. This work was developed in the facilities of Universidad Anahuac Querétaro, the system's signal was emulated in MATLAB. To ensure the lowest percentage of error possible, proposed values were chosen based on their proximity to the original matrix which was calculated with the values of the torque, angular velocity, and angular position. After several tests, an accuracy of 98% was achieved with the A matrix values, this allowed us to correctly estimate linear parameters**

***Keywords*—Kalman filter, mechanical ventilator, parametric identification, dynamical system, state space representation**

I. INTRODUCTION

At the present time, the world is dealing with a new virus named SARS-CoV-2, which is known for causing the coronavirus disease, the first case took place in the city of Wuhan, the capital of Hubei in China. This new virus mainly affects the respiratory system and in some patients the use of an auxiliary ventilator is required.[1]

With the arrival of mechanical ventilators in Mexico, it is essential to highlight the importance of why signal filtering is relevant to help the medical community fight COVID-19. In this case, a Kalman filter prevents noise disruptions, deviations or interferences from similar devices that can lead to triggers on readings that could compromise the patient's health. It works by allowing us to control the angular displacement, angular speed and torque which are the variables used in this specific system. [2]

Since we are working with an auxiliary mechanical ventilation system, in this document authors have applied the simplest version of the Kalman filter as a parametric identification algorithm to estimate linear parameters of the system. It was invented by Rudolf E. Kalman in 1960.

The filter is constructed as a mean squared error minimizer all based on the dynamics of the system and noise observations. [3,4].

Placket, R. *et al.* rediscovered Gauss's theory and exposed the recursive least-squares algorithm which invokes only statistical concepts from the classic linear regression model whereas Kalman's was within a wider context of a state-space model with time-varying values. Nowadays these discoveries are still the core of the Kalman Filter [5].

Kalman filters are a recursive solution to discrete-data linear filtering problems. This filter is a set of mathematical equations that provide an efficient computational (recursive) solution of the least-squares method. The filter is powerful in several aspects since it can make estimations of past, present, and even future states, and it can do so even when the nature of the modeled system is unknown.[6]

With the use of a Kalman filter we can estimate a variable´s value through measured data following two main steps: the first one is to predict the state of the system and then incorporate the new correct measured data, so that we can obtain an optimal estimator. The second one is to understand how signal processing works, since there are always variations that interrupt the main signal, these unwanted variations, commonly called noise, are reduced through filters, such as the Kalman Filter, that allow a better data acquisition procedure [7].

Vignaux, L. *et al.* proposed the non-invasive ventilation programs (NIV) algorithm to evaluate the impact of ventilation algorithms. They estimate that the NIV program can significantly reduce the number of noises caused by respiratory leaks. They found a significant correlation between the magnitude of the leaks and the respiration when the NIV algorithm was not activated.[8]

MIT developed a mechanical ventilator, with the intention to make this technology available for everyone and provide the best safety-focused information on the automation of a manual resuscitator, used in longer-term ventilation. [9]

A student from the MIT used a Kalman filter with the objective of estimate x_k which is the information bearing signal and n_k, which is the additive noise from a general form of describing signals. [10]. He proposed a signal with the following equation:

$$y_k = a_k x_k + n_k \tag{1.1}$$

Where, y_k is the time dependent observed signal, a_k is a gain term, x_k is the information bearing signal and n_k is the additive noise. The main purpose was calculating x_k, the difference between $\hat{x}_k$ and x_k itself is given by the error:

$$(e_k) = (x_k - \hat{x}_k) \tag{1.2}$$

He inferred that the values should both be positive and increasing, the chosen function was the squared error function. The expected value of the error function was stated by the following equation:

$$\text{Cost } function = ((e^2{}_k)) \tag{1.3}$$

The results from the mean square function:

$$\in (t) = E(e^2{}_k) \tag{1.4}$$

He shared that even though the mean square error was intuitive, it was also somewhat heuristic, so they calculated maximum likelihood too. In this case V_k is the associated measurement error, which the author assumes to be a white noise process (w_k) with its correspondent covariance. Through Gaussian distributions the systems errors, were found and the covariances of the two noise models were given by:

$$Q = E(w_k e^T{}_k) \tag{1.5}$$

$$R = E(v_k v^T{}_k) \tag{1.6}$$

From the mean square function *(2.4),* he stated the following equivalence:

$$E(e_k e^T{}_k) = P_k \tag{1.7}$$

Where, P_k is the error covariance matrix at time k. To further understanding and expansion, the equation is now:

$$P_k = E(e_k e^T{}_k) = E[(x_k - \hat{x}_k)(x_k - \hat{x}_k)^T \tag{1.8}$$

By assuming that the prior estimate of $\hat{x}_k$ is called $\hat{x}'_k$ and it was gained by the knowledge of the system and that it is possible to update the equation for a new one that combines the new data with previous estimations, he stated that:

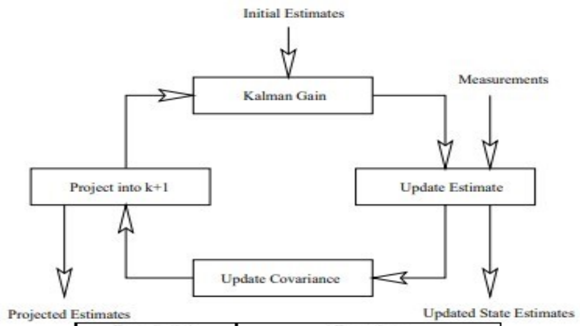
$$\hat{x}_k = \widehat{x'}_k + K_k(z_k - H\widehat{x'}_k) \tag{1.9}$$

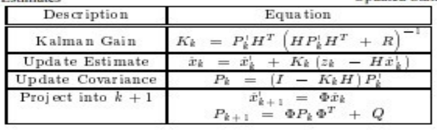
Continuing with substitution he obtained the final error covariance updated equation:

$$P_k = (I - K_k H)P'_k (I - K_k H)^T + K_k R K^T_k) \tag{1.10}$$

Where P'_k is the prior estimate of P_k.

Once this equation was calculated they followed to calculate the matrix. The final recursive algorithm is explained in Figure 1.





Description	Equation
Kalman Gain	$K_k = P'_k H^T \left(H P'_k H^T + R\right)^{-1}$
Update Estimate	$\hat{x}_k = \hat{x}'_k + K_k(z_k - H\hat{x}'_k)$
Update Covariance	$P_k = (I - K_k H)P'_k$
Project into $k+1$	$\hat{x}'_{k+1} = \Phi \hat{x}_k$ $P_{k+1} = \Phi P_k \Phi^T + Q$

Figure 1.- Kalman Filter Recursive Algorithm.

II. Theoretical Background

A Kalman filter has a large amount of advantages, one of those is that it is a recursive process, which means that we can incorporate new data without having to reformulate all the algorithm. [11].

An important condition for an estimation to be considered accurate is that the system must be observable, meaning that the information from the behavior of both the initial and final state is available. Observability for discrete-time systems can be defined as:

$$x(k+1) = Ax(k) + Bu(k) \tag{2.1}$$

$$y(k) = Cx(k) + Du(k) \tag{2.2}$$

It is said that if there is a finite number of time steps (k) so that the information from the input sequence (u) and the output sequence (y) is enough, we can then determine the initial state of the system [12].

State variables and eigenvalues con be obtained through differential equations and measurements made on the system. The Kalman filter provides an estimate with an inevitable level of uncertainty, but with the cost or loss function we are capable of determining how well the filter works [13].

The extended Kalman filter has been applied on non-linear estimations for machine learning applications and dynamic systems. In this case the state approximation is given by the Gaussian variable which can introduce large errors in the true posterior mean and covariance of the transformed variable. The classic framework involves an estimation of the state of a discrete-time nonlinear dynamic system:

$$x_{k+1} = F(x_{k,} v_{k,}) \qquad (2.3)$$

$$y_k = H(x_{k,} n_{k,}) \qquad (2.4)$$

where x_k represents the unobserved state of the system and y_k is the observed signal. The process noise v_k drives the dynamic system, the observation noise is given by n_k and H and F are assumed known. This is the standard method of choice to achieve a recursive maximum likelihood estimation of the initial state [14].

To show and explain our results and analysis process, this document is going to be divided in different sections, starting with an introduction and following on the next chapter our theoretical background, then our description of the applied calculations as well as an explanation of our model, our results and analysis. Finally, we will share our discussion and conclusion of our model.

III. Methodology

Since our aim is to obtain the signals from the angular velocity, torsional torque, and angular position we must analyze their respective formula and equation used to model our system.

$$\frac{da(t)}{dt} = \frac{w(t)}{Ke} \qquad (3.1)$$

Where *Ke* = 3 (gear ratio)

$$\frac{dw(t)}{dt} = -\frac{K\theta Ke}{J\theta}a(t) - \frac{B\theta}{J\theta}w(t) + \frac{1}{J\theta}T(t) \qquad (3.2)$$

Where:

$K\theta = 0.01(angular\ stiffness)\ \ (N/m)$
$J\theta = 25(torsional\ moment\ of\ inertia)\ (Ns^2)/m$
$B\theta = 2(viscous\ damping\ coefficent)\ (Ns)/m$

$$\frac{dT(t)}{dt} = -\frac{Km2}{Lkm1}w(t) - \frac{R}{L}T(t) + \frac{1}{LKm1}V(t) \qquad (3.3)$$

Where:

$Km1 = 100(speed - voltage\ ratio)$ [Vs/rad]
$Km2 = 2.5(torque\ current\ ratio)$ [A/Nm]
$R = 5(motor's\ resistance)$ [Ω]
$L = 0.82(motor's\ inductancy)$ [Ωs]

With these initial values we generated the construction of the system in state space representation.

$$\frac{d}{dt}\begin{bmatrix} a(t) \\ w(t) \\ T(t) \end{bmatrix} = \begin{bmatrix} 0 & 1/Ke & 0 \\ -\dfrac{K\vartheta Ke}{J\vartheta} & -\dfrac{B\vartheta}{J\vartheta} & \dfrac{1}{J\vartheta} \\ 0 & -\dfrac{Km2}{LKm1} & -\dfrac{R}{L} \end{bmatrix}\begin{bmatrix} a(t) \\ w(t) \\ T(t) \end{bmatrix} + \begin{bmatrix} 0 \\ 0 \\ \dfrac{1}{LKm1} \end{bmatrix}[V(t)]$$

Figure 2.- space of states representation

Then we proceeded with the substitution of values.

$$\frac{d}{dt}\begin{bmatrix} a(t) \\ w(t) \\ T(t) \end{bmatrix} = \begin{bmatrix} 0 & 1/3 & 0 \\ -\dfrac{3}{2500} & -\dfrac{2}{25} & \dfrac{1}{25} \\ 0 & -\dfrac{2.5}{82} & -\dfrac{5}{0.82} \end{bmatrix}\begin{bmatrix} a(t) \\ w(t) \\ T(t) \end{bmatrix} + \begin{bmatrix} 0 \\ 0 \\ \dfrac{1}{82} \end{bmatrix}[V(t)]$$

Figure 3.- values of the space of states representation

After we obtained the differential equations of the system and represented the system in a state-space representation, we designed an algorithm in MATLAB to get the system's output. First, we designed the initial matrix A with experimental results shown in Figure 3.

The system was modeled in its state-space representation (*see Figure 4*) in order to obtain the angular position, torsional torque and angular velocity from the system. To obtain them we used a parametric identification algorithm with one input and multiple outputs programed in MATLAB.

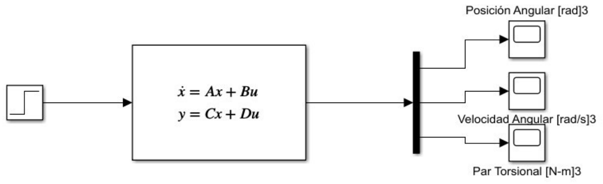


Figure 4.- Simulation in space of states

We followed by testing the algorithm with one volt as an input, the results are shown on the following graphs (see from Figure 4 to Figure7). Each parameter corresponds to every output from the system, showed in the previous programmed algorithm.

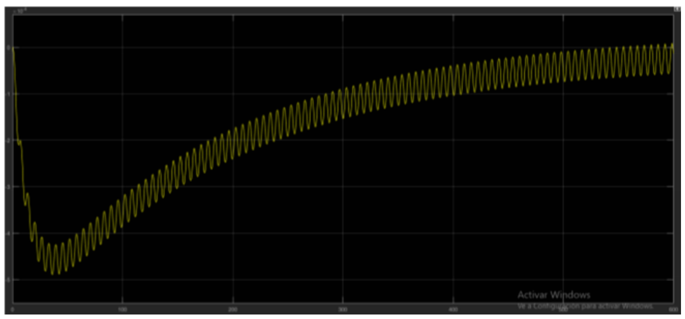


Figure 5.- Angular position

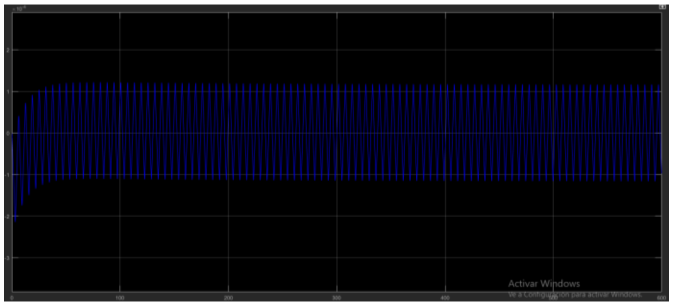


Figure 6.- Angular Velocity

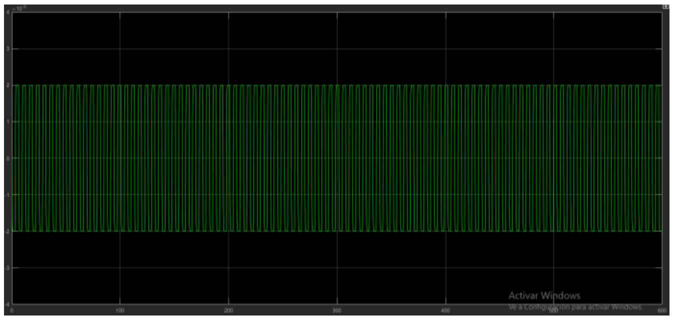


Figure 7.- Torsional torque

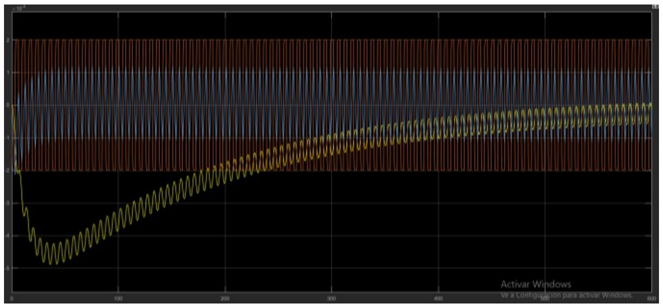


Figure 8.- Torsional Torque, angular position, and angular Velocity

Now that we know exactly how the system behaves and how the parameters react to the original values and its arrangement inside a matrix, we can work on our proposal for a new matrix A, and then determine if these values can actually work for our system by calculating the square error of our results compared to the originals.

IV. Results

We programmed an intelligent algorithm based in the differential equations that compose the initial matrix A. The algorithm had the responsibility to analyzes all the differential equations and gives us the values that were closer to the initial matrix A.

Before we iterated the algorithm several times, we obtain very similar parameters as we can see in Figures 8, 9 and 10. The problem was that the angular velocity values were inverted as we can see in Figure 10.

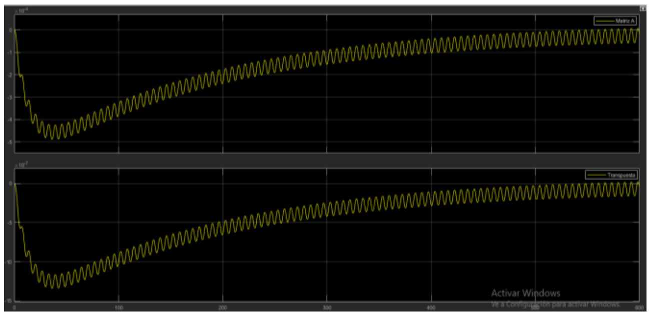


Figure 9.- Angular position

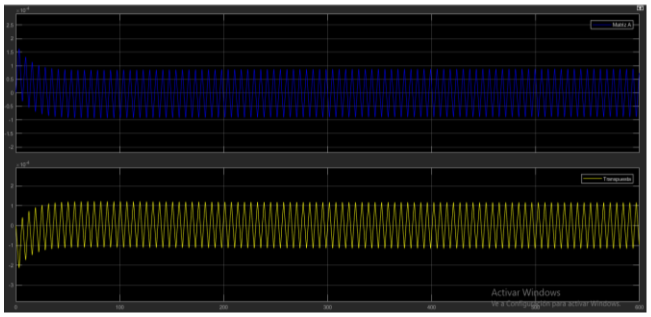


Figure 10.- Angular velocity

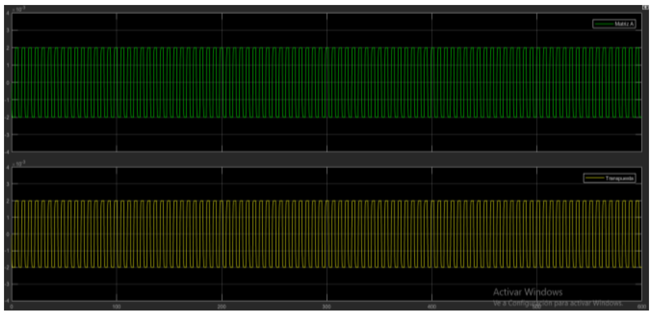


Figure 11.- Torsional Torque

Even though the algorithm were transposing the angular velocity values we decided to implement in order to analyze it performance. When we were analyzing the performance of the algorithm using the estimated matrix A, we observed that the angular velocity error did not decrease as we can see in Figure 12.

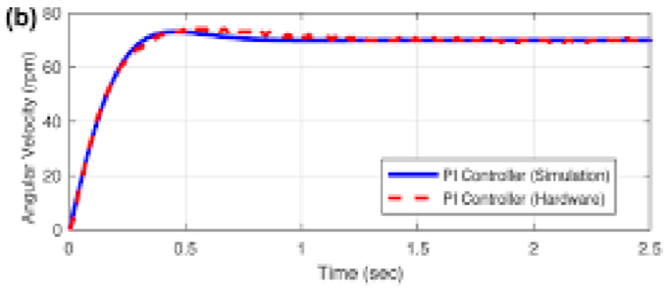


Figure 12.- Angular Velocity Error

Then we decided to implement the same algorithm, but this time providing the angular velocity values transpose. After we iterate again the algorithm, we obtained correctly the estimated initial matrix A. Furthermore, the algorithm performs in the way that we wanted.

Table 1.- Matrix comparicion

Initial matrix A	*Initial matrix A with ω^T*
$\begin{bmatrix} 0 & \frac{1}{Ke} & 0 \\ -\frac{K\theta Ke}{J\theta} & -\frac{B\theta}{J\theta} & \frac{1}{J\theta} \\ 0 & -\frac{Km2}{Lkm1} & -\frac{R}{L} \end{bmatrix}$	$\begin{bmatrix} 0 & \frac{1}{Ke} & 0 \\ (-\frac{K\theta Ke}{J\theta})T & (-\frac{B\theta}{J\theta})T & (\frac{1}{J\theta})T \\ 0 & -\frac{Km2}{Lkm1} & -\frac{R}{L} \end{bmatrix}$

Now the algorithm reduces the angular velocity error as we can see in the Figure 13.

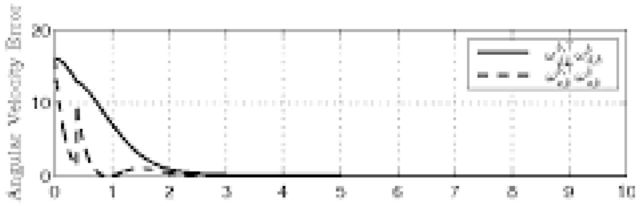


Figure 13.- Angular Velocity Error

From the best algorithm we encounter the following values for each linear parameter:

- Ke = 2.8 (gear ratio)
- $K\theta = 0.01(angular\ stifness)\ [N/m]$
- $J\theta = 25$ (*torsional moment of inertia*) [Ns^2]/[m]
- $B\theta = 2$(*viscous damping coefficient*) [Ns]/[m]
- Km1 = 100 (*speed-voltage ratio*) [vs] /[rad]
- Km2 = 2.25(*torque – current ratio*) [A] /[Nm]
- $R = 5(motor's\ resistance)\ [\Omega]$
- $L = 0.8(motor's\ inductancy)\ [\Omega s]$

Then we proceeded to replace the initial matrix A values to the estimated matrix A values.

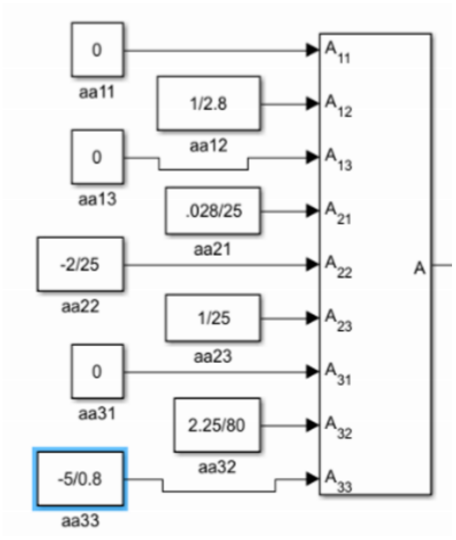


Figure 14.- Values of the Estimated matrix A

Estimated matrix A:

$$A(nT) = \begin{bmatrix} 9x10^{-19} & 0.357 & 2.25x10^{-21} \\ -1.11x10^{-3} & -7.2x10^{-2} & \frac{1}{25} \\ 0 & -2.74x10^{-2} & -6.09 \end{bmatrix}$$

Figure 15.- Estimated matrix A

We obtained the following system by approximating the values from the estimated matrix A:

$$\frac{d}{dt}\begin{bmatrix} a(t) \\ w(t) \\ T(t) \end{bmatrix} = \begin{bmatrix} 0 & 1/2.8 & 0 \\ -\frac{7}{6250} & -\frac{2}{25} & \frac{1}{25} \\ 0 & -\frac{2.25}{80} & -\frac{5}{0.8} \end{bmatrix} \begin{bmatrix} a(t) \\ w(t) \\ T(t) \end{bmatrix} + \begin{bmatrix} 0 \\ 0 \\ \frac{1}{80} \end{bmatrix} [V(t)]$$

Figure 16.- final system (Ax+Bu)

In order to determine our error, we compared our matrixes. Since we wanted this error to be as minimum as possible, we will later equal it to cero, to proceed with our calculations. *(See Figure 13).*

$$(nT) = (nT) - X_{estimate(nT)} \tag{4.1}$$

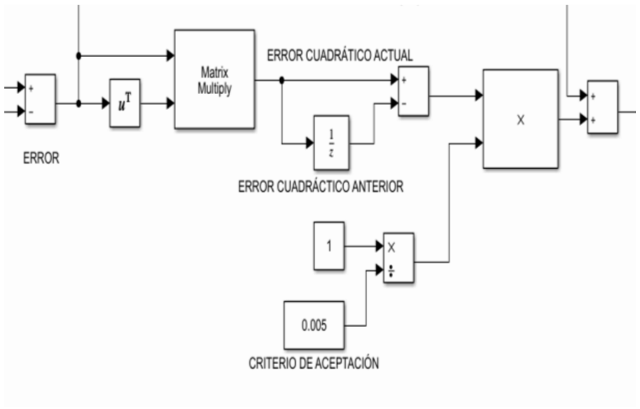


Figure 17.- Obtaining our square error

To calculate the error, we needed to operate different matrix equations, so we started by defining the real and estimate output.

where:

$$(nT) = Real\ output \tag{4.2}$$
$$X_{estimate(nT)} = Estimated\ output$$

Then we had that the error equation is squared:

$$E^2_{3X3(nT)} = E(nT)_{3x1} * E^T_{3x1(nT)} \tag{4.3}$$

This derives from the error in respect to the variable of interest, this will help us later on making it easier for us to obtain our variable of interest.

$$\frac{dE^2(nT)}{dA(nT)} = \frac{1}{a}\left(E^2(nT) - E^2(nT-T)\right) + A(nT-T) - A(nT) \tag{4.4}$$

The derivative of the error was equalized to zero, since we needed the error to be as low as possible, in order to minimize it and obtain our variable of interest.

$$A(nT) = \frac{1}{a}(E^2(nT) - E^2(nT - T) + A(nT - T) \qquad (4.5)$$

Now that we have the variable that we needed to find, we then proceeded to calculate the estimated output, which is:

$$x_{estimate} = (nT) + Bu \qquad (4.6)$$

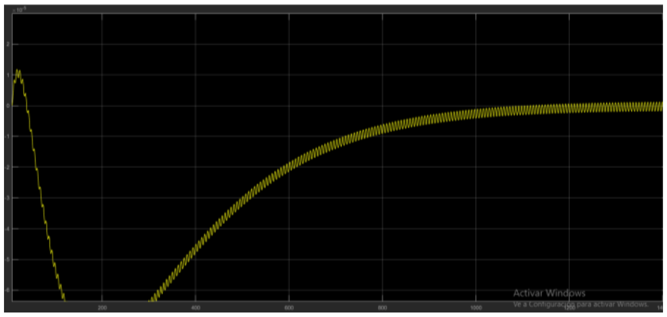


Figure 18.- Angular position error

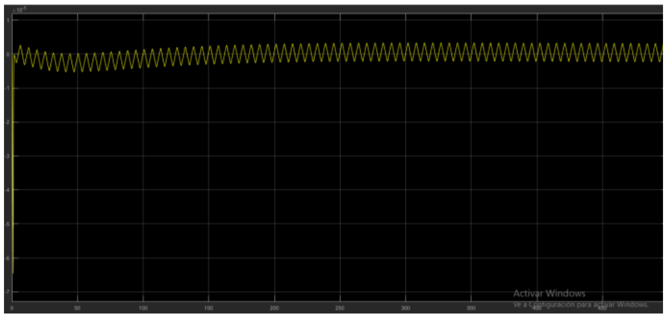


Figure 19.- Angular velocity error

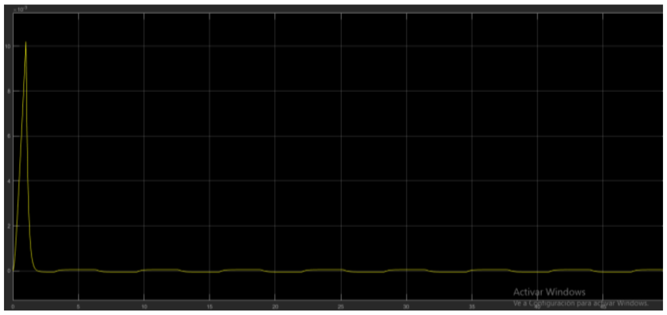


Figure 20.- Torsional Torque error

As we can see from our graphics, our error is very low, since it is variating from ranges of x10^{-5}, showing that our estimated values were indeed correct, and they are very accurate in terms of the results obtained from comparing both matrixes.

Now that we obtained the error for each value, we proceeded to graph our three main values: torsional torque, angular velocity, and angular position, to see how much they resemble the original graphs we obtained.

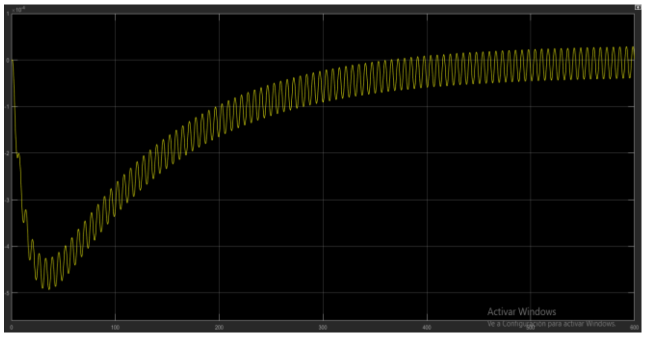


Figure 21.- Estimated angular position

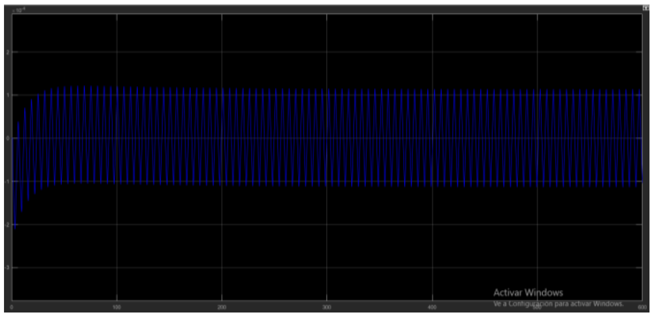


Figure 22.- Estimated angular velocity

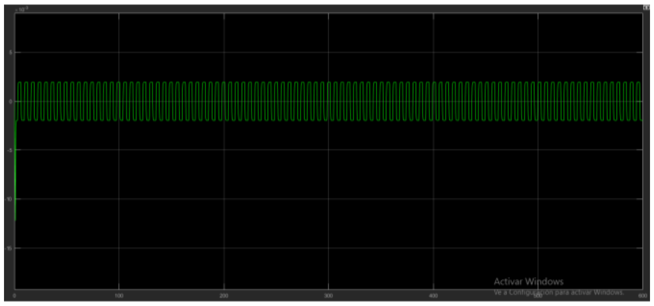


Figure 23.- Estimated torsional torque

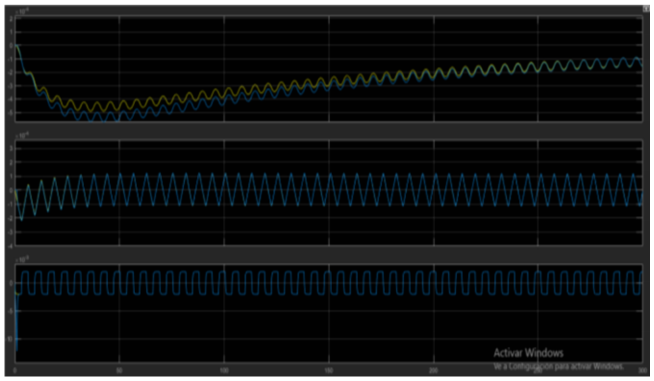


Figure 24.- Comparison of actual with estimated output

The graphics for the original values and the estimation are almost identical as we can see in Figure 24. We obtain an accuracy of 98% of similarity between both matrixes.

Now in order to obtain our Eigenvalues from the initial matrix A, we used MATLAB to create an input for every single value that the matrix A had, and then generate these eigenvalues.

Eigenvalues:

$$A=\begin{bmatrix} 0 & \frac{1}{k_e} & 0 \\ -\frac{K\emptyset K_e}{J\emptyset} & -\frac{B\emptyset}{J\emptyset} & \frac{1}{J\emptyset} \\ 0 & -\frac{JK_{m2}}{LK_{m1}} & -\frac{R}{L} \end{bmatrix} + \begin{bmatrix} \lambda & 0 & 0 \\ 0 & \lambda & 0 \\ 0 & 0 & \lambda \end{bmatrix}$$

$$det\begin{bmatrix} \lambda & -\frac{1}{3} & 0 \\ \frac{(0.01)(3)}{25} & \lambda+\frac{2}{25} & -\frac{1}{25} \\ 0 & \frac{2.5}{(0.82)(2.5)} & \lambda+\frac{5}{0.82} \end{bmatrix}$$

$$=\lambda\left[\left(\lambda+\frac{2}{25}\right)\left(\lambda+\frac{5}{0.82}\right)+\left(\frac{1}{25}\right)\left(\frac{2.5}{(0.8)(2.5)}\right)\right] + -\frac{1}{3}\left[\left(\frac{(0.01)(3)}{25}\right)\left(\lambda+\frac{5}{0.82}\right)\right]$$

$$=\lambda^3+6.17756\lambda^2+0.792283\lambda-0.002328=0$$

- λ_1= -6.0844
- λ_2= -0.0833094
- λ_3= -0.0048076

For the estimated matrix A, we obtained the following Eigenvalues to compare them with the original ones:

$$det\begin{bmatrix} 0 & 1/2.8 & 0 \\ -\frac{7}{6250} & -\frac{2}{25} & \frac{1}{25} \\ 0 & -\frac{2.25}{80} & -\frac{5}{0.8} \end{bmatrix} + \begin{bmatrix} \lambda & 0 & 0 \\ 0 & \lambda & 0 \\ 0 & 0 & \lambda \end{bmatrix}$$

$$A=\begin{bmatrix} \lambda & -1/2.8 & 0 \\ \frac{7}{6250} & \lambda+\frac{2}{25} & -\frac{1}{25} \\ 0 & \frac{2.25}{80} & \lambda+\frac{5}{0.8} \end{bmatrix}$$

$$=\lambda\left(\left(\lambda+\frac{2}{25}\right)\left(\lambda+\frac{5}{0.8}\right)-\left(-\frac{1}{25}\right)\left(\frac{2.25}{80}\right)\right)+\frac{1}{2.8}\left(-\frac{7}{6250}\right)\left(\lambda+\frac{5}{0.8}\right)$$

$$=\lambda^3+6.33\lambda^2+0.5004\lambda+0.0025=0$$

- $\lambda_1=-6.24982$
- $\lambda_2=-0.0748372$
- $\lambda_3=-0.00534509$

V. CONCLUSIONS AND DISCUSSIONS

With a simple algorithm programmed in MATLAB and the use of differential equations we correctly applied the state space method which allowed us to obtain the best parameter estimation possible as we can see from figures 21 to 24, as well as the lowest error regarding the parameters estimation with an accuracy of 98% shown in figures 14 to 17.

As we can see, our eigenvalues are very similar to the ones obtained in the original matrix, with variations of 0.001 and a minimum range of error. Based on the results obtained we can determine our filter as successful and efficient, since it does approximate to the original proposed matrix and the difference between them both is minimal. In terms of the filter, we conclude that the output of a differential equation can be in fact estimated by discretizing the equation in finite differences and in order to minimize the associated error. The mean square error method is effective to reduce differences between the final and estimated input in order for the algorithm to converge by only having to adjust the parameters of interest.

Compared to other parametric identification algorithms such as the non-invasive ventilation algorithms (NIV), MIT mechanical ventilator system, and decision support system for mechanical ventilation (DSS) our intelligent algorithm is computationally less expensive.

ACKNOWLEDGEMENT

Authors thanks MIT, SSC group, CIDESI and all those people that have supported the virtual mechanical ventilators CAD design and open loop simulations, the input data for this work was obtained by considering the open source mechanical ventilator information without any modification documented or considered on.[15]

CLARIFICATION NOTES

It is important to clarify that the all information used in this work does not include any modification from the open source design available for the mechanical ventilator, neither on hardware not on software [6], because this work scope just has the educational intention to verify how this kind of medical devices reacts when use a Kalman filter; therefore, any authors rights from were not affected.

REFERENCES

[1] Organización Mundial de la Salud (2020). Nuevo Coronavirus China Retrieved 12 January, 2020 from https://www.who.int/csr/don/12january-2020-novel-coronavirus-china/es/

[2] Navarro, M. (2020). El Coronavirus 19 (COVID 19) en México. Retrieved from https://www.forbes.com.mx/actualidad-unops-errorsistemas-salud-mercado/

[3] Pei, Y. Fussel, D. Biswas, S. Pingali, K. (2019). An elementary introduction to Kalman filters. Retrieved 17 March,2020 from https://arxiv.org/pdf/1710.04055.pdf.

[4] Munuera, M. (2018) Filtro de Kalman y sus aplicaciones. Retrieved 16 March 2020 from http://diposit.ub.edu/dspace/bitstream/2445/127417/2/memoria.pdf

[5] Pollock, S. (n.d). Recursive estimation and the Kalman filter. Retrieved 16 March 2020 from https://www.le.ac.uk/users/dsgp1/COURSES/MESOMET/ECMETXT/recurse.pdf

[6] Welch, G and Bishop G (1997). An Introduction to the Kalman Filter. Retrived 17 September 1997 from http://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.336.5576&rep=rep1&type=pdf

[7] Grewal, M. Andrews, A. (2014) Kalman Filtering: Theory and Practice with MATLAB. US: JohnWiley & Sons.

[8] Vignaux, L. Tassaux, D. Carteaux G. Roeseler J. Piquilloud G. Brochard L. and Jolliet P. (2010). Rendimiento de algoritmos de ventilación no invasiva en ventiladores de UCI durante la presión de soporte: un estudio clínico. Retrieved 6 August, from https://pubmed.ncbi.nlm.nih.gov/20689921/

[9] MIT (2020). MIT Emergency Ventilator Project Retrieved from https://emergency-vent.mit.edu

[10] Lacey, T (1998). Tutorial: The Kalman Filter Retrieved from http://web.mit.edu/kirtley/kirtley/binlustuff/literature/control/Kalman%20filter.pdf

[11] Haugen, F. (n.d) State estimation with Kalman Filter. Retrieved 17 March,2020 from http://techteach.no/fag/seky3322/0708/kalmanfilter/kalmanfilter.pdf

[12] Mohan, S. Naik, N. Gemson, M. (2015). Introduction to the Kalman filter and tuning its statistics for near optimal estimates and Cramer Rao Bound. Retrieved 17 March 2020 from https://arxiv.org/pdf/1503.04313.pdf

[13] Cao, Y. (2016) The application of finite-difference extended Kalman filter in GPS speed measurement. China: ICMMCT, from https://doi.org/10.2991/icmmct-16.2016.341

[14] Mohan, S. Naik, N. Gemson, M. (2015). Introduction to the Kalman filter and tuning its statistics for near optimal estimates and Cramer Rao Bound. Retrieved 17 March 2020 from https://arxiv.org/pdf/1503.04313.pdf

[15] CONACYT, Dirección General, Comunicado 163/20 https://www.conacyt.gob.mx/index.php/glosario-de-terminos-sni/109-comunicados/1284-com-163-2020