PARTICLE SWARM OPTIMIZATION OF AN EXTENDED KALMAN FILTER FOR SPEED ESTIMATION OF AN INDUCTION MOTOR

THUẬT TOÁN TỐI ƯU BẦY ĐÀN CỦA BỘ LỌC KALMAN MỞ RỘNG ĐỂ ƯỚC LƯỢNG TỐC ĐỘ CỦA ĐỘNG CƠ CẢM ỨNG

Dao Dinh Thien, Pham Duy An

University of Science and Technology of Hanoi, Vietnam Academy of Science and Technology

Ngày nhận bài: 11/12/2024, Ngày chấp nhận đăng: 28/12/2024, Phản biện: TS. Hoàng Trung Kiên

Abstract:

Induction motors have always been and will be the forefront of the industrial process. From construction to automated and electric vehicles, induction motors are the first choice. The rotor speed is an essential property in the efficiency of an induction motor which can be measured directly through an optical encoder which is placed on the motor shaft. However, the system cost, volume and weight of the motor are increased, moreover, the reliability and efficiency of the sensor are jeopardized in harsh environments. This paper proposes a method using an Extended Kalman Filter (EKF) to estimate the rotor speed with or without a sensor while reducing the noise created by a harsh environment. Furthermore, with the implementation of the Particle Swarm Optimization algorithm (PSO), a more accurate estimation has been achieved. Simulation in MATLAB Simulink has been done to further emphasize on the efficiency of the EKF-PSO framework.

Keywords: Extended Kalman Filtering, Particle Swarm Optimization, Induction Motor.

Tóm tắt:

Động cơ cảm ứng luôn đóng vai trò tiên phong trong quá trình công nghiệp hóa. Từ xây dựng đến các phương tiện tự động và xe điện, động cơ cảm ứng luôn là sự lựa chọn hàng đầu. Tốc độ rotor là một thuộc tính thiết yếu ảnh hưởng đến hiệu suất của động cơ cảm ứng, có thể được đo trực tiếp bằng cách sử dụng bộ mã hóa quang học gắn trên trục động cơ. Tuy nhiên, việc sử dụng cảm biến này làm tăng chi phí, kích thước và trọng lượng của động cơ, đồng thời độ tin cậy và hiệu quả của cảm biến có thể bị ảnh hưởng trong môi trường khắc nghiệt. Bài báo này đề xuất một phương pháp sử dụng Bộ lọc Kalman mở rộng (Extended Kalman Filter - EKF) để ước tính tốc độ rotor với hoặc không cần cảm biến, đồng thời giảm nhiễu gây ra bởi môi trường khắc nghiệt. Hơn nữa, với việc triển khai thuật toán Tối ưu hóa Bầy đàn Hạt (Particle Swarm Optimization - PSO), độ chính xác trong ước tính được cải thiện đáng kể. Mô phỏng trong MATLAB Simulink đã được thực hiện để nhấn mạnh thêm hiệu quả của khung EKF-PSO.

Từ khóa: Bộ lọc Kalman mở rộng, Tối ưu hóa Bầy đàn hạt, Động cơ cảm ứng.

1. INTRODUCTION

The induction motor (IM) was invented in the end of the 19th century and more than a century later, it still remains at the forefront of multiple industrial processes. Due to its simplicity, versatility, reliability and economical nature [1][2], it is still the first choice among in numerous industrial processes [3] and developing industries [21]. Rotor speed is an essential parameter of an IM and has been the subject of numerous articles [5][6]. However, the most obvious method to measure the speed of a rotor is to mount a sensor on the motor shaft which can be costly and can increase the weight and the volume of the induction motor [7]. Moreover, in rough environments, the measurements could be inaccurate due to the noise. In this paper, we will examine the use of a Kalman Filter in order to reduce the noise and estimate the rotor speed. In literature, the Kalman Filter (KF) has been a hot topic to be used to reduce the environmental noise [8] [9]. But due to the non-linearity nature of the dynamic model, the Extended Kalman Filter (EKF) will be used instead of a KF.

An EKF is a stochastic observer for non-linear systems [10][11]. It is especially efficient when dealing with multivariate non-linear systems as it uses linear approximations to deal with this non-linearity. Furthermore, the EKF can accurately model a system that is burden by system and measurement noises. Indeed, these disturbances are effectively modelled in the filter through covariance matrices dubbed the system noise and measurement noise covariance matrices. The values of these matrices are crucial to the accuracy of the system. These values are however not determined by using a formula and are not known. Therefore, they must be tuned accurately in order to achieve an accurate estimation.

Indeed, in the past, the covariance matrix could be tuned either manually [12] and the process could be extremely laborious or by applying mathematically tuning methods [13] [14][15]. So, in order to reduce the long and rigorous process of tuning the EKF, methods such as the Ant Colony Optimization [16] or the Genetic Algorithm [17] have been used. But we will be using the Particle Swarm Optimization (PS.O) [18] due to its simplicity, its relatively easy implementation and its fast convergence.

Developed in 1995 by James Kennedy and Russel C. Eberhart, the PSO is based on a simplified social model. It is used in image processing [19], in air route optimization [20], in acoustic filters optimization [21]. However, we will be using the PSO in order to optimize the system noise covariance matrix and the measurement noise matrix. By optimizing these matrices, we will be able to have an accurate estimation of the rotor speed despite of the environmental and measurement noise. PSO have been used to estimate the rotor speed [22][23] offline but this paper will propose a method in which the stator current will be estimated in order to estimate the rotor speed.

The goal of the paper is to show the efficiency of the EKF to accurately estimate a measurement despite the noises that burden the system. In addition, it will also show the improvement that the P.S.O will bring to the EKF model in a model in which a speed sensor is used [23] and improve upon it in a model in which there is no speed sensor.

The rest of the paper will be divided as follows: Section 2 will present the mathematical on which the EKF is based. The Simulink model of the AC induction motor will then be described in section 3. Afterwards, section 4 will cover the core concept of PSO. Section 5 will depict the implementation of PSO in the EKF model. Simulation results will be analysed in section 6 and finally conclusions are presented in section 7.

2. EXTENDED KALMAN FILTER

The extended Kalman Filter is in essence an amelioration of the Linear Kalman Filter. The Linear Kalman Filter is only limited to linear problems, however most real-life problems are in fact non-linear. The Extended Kalman Filter uses linear approximation to study non-linear dynamic systems. Indeed, this filter is developed upon a first-order Taylor Series approximation of the state transition and the measurement function. The EKF is an algorithm where we constantly extrapolate or "predict" our system state and the uncertainty of the next step as shown in Figure 1. And once the measurements are received, the EKF updates or "corrects" the predictions and we repeat these procedures until the end of the sample period. It is governed by the 5 following equations.

The state extrapolation equation:

$$\widehat{x}_{n+1,n} = f(\widehat{x}_{n,n}, u_n) \qquad (1)$$

Whereas, *f()* is the state transition function based on the state dynamics, is the control input.

The covariance extrapolation equation:

$$P_{n+1,n} = \frac{\delta f}{\delta x} P_{n,n} (\frac{\delta f}{\delta x})^T + Q_n$$

With $\frac{\delta f}{\delta x}$ as the partial derivative of the state transition function obtained per first order Taylor series expansion. And Q_n is the system noise covariance matrix.

The Kalman gain equation:

$$K_n = P_{n,n-1} (\frac{\delta h}{\delta x})^T \times (\frac{\delta h}{\delta x} P_{n,n-1} \left(\frac{\delta h}{\delta x}\right)^T + R_n)^{-1}$$

$\frac{\delta_h}{\delta_x}$ is the partial derivation of the observation function obtained by using a first order Taylor series expansion and is the measurement noise covariance matrix.

The state update equation:

$$\widehat{x}_{n,n} = \widehat{x}_{n,n-1} + K_n(z_n - h(\widehat{x}_{n,n-1}))$$

$\widehat{x}_{n,n-1}$ is the priori state prediction vector of $\widehat{x}_{n,n}$, Z_n is the measurement and *h()* is the observation function.

The covariance update equation:

$$P_{n,n} = (I - K_n \frac{\delta h}{\delta x}) P_{n,n-1}$$

$P_{n,n-1}$ is the priori prediction vector prediction of $P_{n,n}$.

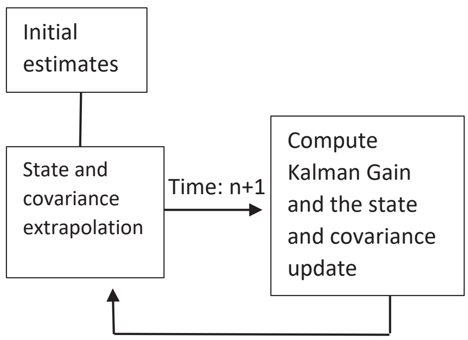

Figure 1. Simplified diagram of the EKF

$\frac{\delta f}{\delta x}$ will take the form of a Jacobian matrix like so:

$$\frac{\delta f}{\delta x} = \begin{pmatrix} \frac{\delta f_1}{\delta x_1} & \cdots & \frac{\delta f_1}{\delta x_n} \\ \vdots & \ddots & \vdots \\ \frac{\delta f_m}{\delta x_1} & \cdots & \frac{\delta f_m}{\delta x_n} \end{pmatrix}$$

As the Extended Kalman Filter can model a dynamic system despite of the noises and uncertainty, the Q_n system noise covariance and the R_n measurement noise covariance matrix which are the mathematical modelling of the noise are extremely important. All covariance matrix takes the form of $E = (ee^T)$. So, Q_n= E($W_n W_n^T$) and $R_n = (V_n V_n^T)$ with W_n being the process noise and V_n is the measurement error.

3. Induction motor model and discretization

The induction motor model has been modelled in MATLAB, Simulink in the stationary reference frame αβ. The stator and rotor fluxes and the stator and rotor currents have been modelled. Moreover, system noise and measurement noise have been added to the system as seen in the subsystem "stator current" in Figure 2, in order to have an accurate simulation. From the simulation, only stator currents and the voltages would

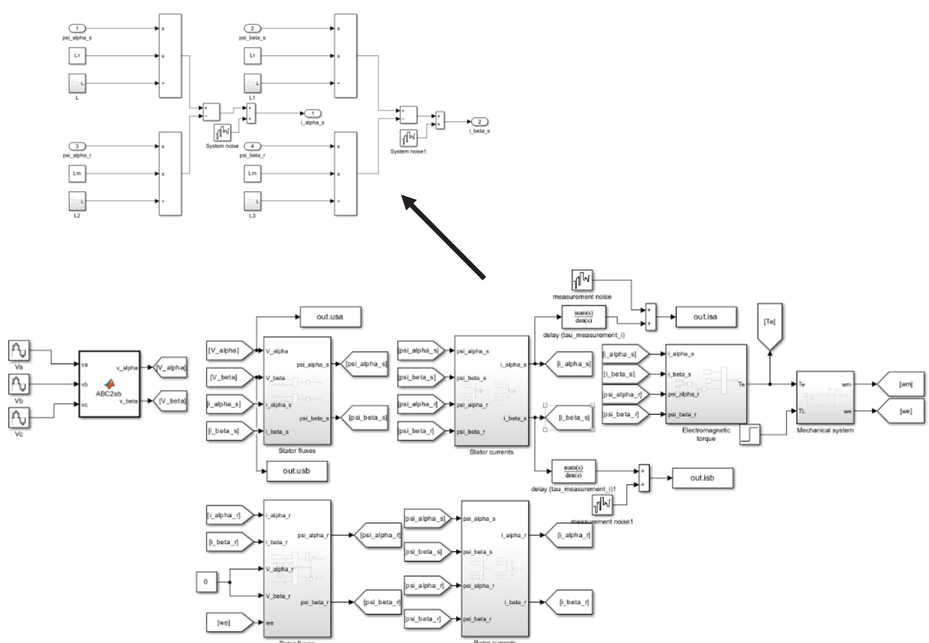

Figure 2. Simulink model of Induction motor

be transfer back to MATLAB to be used as values for the measurement input matrix and the control input matrix. The state variables, the control input and the measurements input are therefore the following:

$$x(t) = [x_1\ x_2\ x_3\ x_4\ x_5]^T$$
$$= [i_{\alpha s}\ i_{\beta s}\ \psi_{\alpha r}\ \psi_{\beta r}\ \omega]^T$$
$$u_n = [v_{\alpha s}\ v_{\beta s}]^T$$
$$z_n = [i_{\alpha s}\ i_{\beta s}]^T$$

This would enable us to estimate the speed and compare it with the real system speed (plagued by noise) and the theoretical physical speed which is not burden by noise at all. Once the state variables, the control input and the measurement input are selected, the state transition function and its Jacobian matrix can now be constructed. The state transition function is therefore the following:

$$f(x_{n,n}, u_n) = [x_{1,n+1}\ x_{2,n+1}\ x_{3,n+1}\ x_{4,n+1}\ x_{5,n+1}]^T$$

$$f(x_{n,n}, u_n) = \begin{bmatrix} a_0 x_{1,n} + a_1 x_{3,n} + a_2 x_{4,n} x_{5,n} + a_3 u_{1,n} \\ a_0 x_{2,n} + a_1 x_{4,n} - a_2 x_{3,n} x_{5,n} + a_3 u_{2,n} \\ a_4 x_{1,n} + a_5 x_{3,n} - a_6 x_{4,n} x_{5,n} \\ a_4 x_{2,n} + a_5 x_{4,n} + a_6 x_{3,n} x_{5,n} \\ a_7 x_{5,n} + a_8 (x_{2,n} x_{3,n} - x_{1,n} x_{4,n}) - a_9 T_L \end{bmatrix}$$

Once the state transition function is achieved, its Jacobian matrix is then derived:

$$\frac{\delta f}{\delta x} = \begin{bmatrix} a_0 & 0 & a_1 & a_2 x_{5,n} & a_2 x_{4,n} \\ 0 & a_0 & -a_2 x_{5,n} & a_1 & -a_2 x_{3,n} \\ a_4 & 0 & a_5 & -a_6 x_{5,n} & -a_6 x_{4,n} \\ 0 & a_4 & a_6 x_{5,n} & a_5 & a_6 x_{3,n} \\ -a_8 x_{4,n} & a_8 x_{3,n} & a_8 x_{2,n} & -a_8 x_{1,n} & a_7 \end{bmatrix}$$

Since, *h()* is the observation function which links the state variables with the measurements. So, the measurement matrix is the following:

$$z_n = h(x_{n,n})$$

With, $h(x_{n,n-1}) = \begin{bmatrix} x_{1,n-1} \\ x_{2,n-1} \end{bmatrix}$

The discretization of the system's dynamics is now finished and the system is given by the following discrete general system:

$$\begin{cases} x(t+1) = f(x(t), u(t)) \\ z(t) = h(x(t)) \end{cases}$$

4. Particle Swarm Optimization

Particle swarm optimization developed by Eberhart and Kennedy in the late 20th century is an optimization algorithm. Derived from the study of birds' foraging behaviour, it is based on the behaviour of birds when scavenging for food. They don't know the location of the food and only know the distance to the food and the bird whose the closest to the food so they converge on said bird. The PSO algorithm aims to replicate this social phenomenon in order to search for an optimized solution for a problem [24]. The algorithm first initialize a population or a group of particles which is called a swarm.

The number of particles can vary and depending on the size of the swarm. On one hand, when the swarm is large, it will provide a better coverage of the search space, enhance its exploration capability and reduce the risk of getting trapped in a local optima. However, it may result in a slower convergence since there are more particles to evaluate. On the other hand, a smaller swarm size provides a faster convergence however it may not explore the search space adequately and could increase the risk of premature convergence to local optima. Each particle in the swarm of size N is designated by a position x_i and a velocity v_i. Each particle represents a potential solution. The best solution of each particle achieved during the entire optimization process is named p_{best} (local best position) while the best solution to the problem of any particle is named g_{best} (global best position). The local best position and the global best position are determined through a fitness function. The process is depicted in Figure 3.

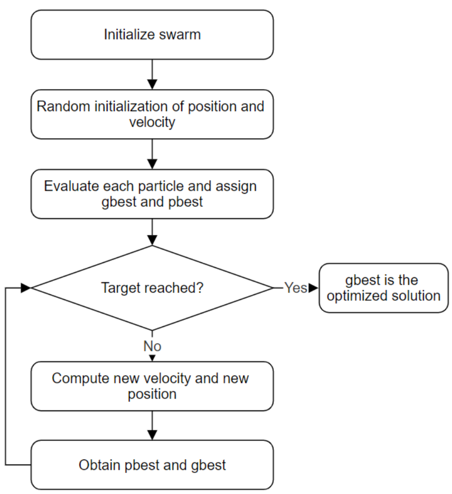

Figure 3. PSO algorithm flowchart

The position and the velocity of each particle are updated at each iteration by the following equations:

$$v(t+1) = w \cdot v_i(t) + c_1 \cdot r_1 \cdot (p_{i,best} - x_i(t)) + c_2 \cdot r_2 \cdot (g_{best} - x_i(t))$$

$$x_i(t+1) = x_i(t) + v(t+1)$$

Whereas, r_1 and r_2 are random coefficients drawn from an uniform distribution of range [0,1]. w is the inertia weight which controls the effects of a prior velocity on its current velocity. The higher the inertia weight, the deeper the exploration. Finally, c_1 and c_2 are the cognitive coefficient and the social coefficient respectively. While a high cognitive coefficient can lead to a more localized search, a high social coefficient will lead to a more broaden exploration of the search space.

5. Implementation of PSO into EKF

Q_n and R_n are extremely important in the operation of the EKF however, since they represent noises which corresponding properties are unknown, determining them is extremely difficult. In the past, there were techniques to determine these parameters however, these techniques are often laborious and highly time consuming. So, in order to reduce the time required to determine these parameters and to avoid unnecessary errors, optimization techniques such as the genetic algorithm has been implemented into the EKF.

In this section, a PSO algorithm will be applied to the EKF in order to tune Q_n and R_n parameters which is shown below:

$$Q_n = \begin{bmatrix} q_i & 0 & 0 & 0 & 0 \\ 0 & q_i & 0 & 0 & 0 \\ 0 & 0 & q_\psi & 0 & 0 \\ 0 & 0 & 0 & q_\psi & 0 \\ 0 & 0 & 0 & 0 & q_\omega \end{bmatrix} and\ R_n = \begin{bmatrix} r_i & 0 \\ 0 & r_i \end{bmatrix}$$

The four parameters of the system noise covariance matrix and the measurement noise matrix will be considered as free variable to be tuned by PSO. The fitness function used to evaluate the parameters will be based on a mean squared error (MSE) criteria in order to estimate the speed. The mean squared error is determined by the following formula:

$$MSE = \frac{1}{n}\sum_{i=1}^{n}(Y_i - \widehat{Y}_i)^2$$

In the following section, two methods of implementing PSO into EKF will be presented. Firstly, the fitness function will compare the measured speed and the EKF estimated speed and calculate the MSE between them. This method will be done in theoretically offline. Nevertheless, it is still a good demonstration of the usefulness of the implementation of PSO into EKF. The sensor less method consists of comparing the stator current β with its EKF counterpart and once again comparing them to obtain the MSE. This method, on the other hand, could theoretically be executed in real time hence rendering the need for a speed sensor obsolete.

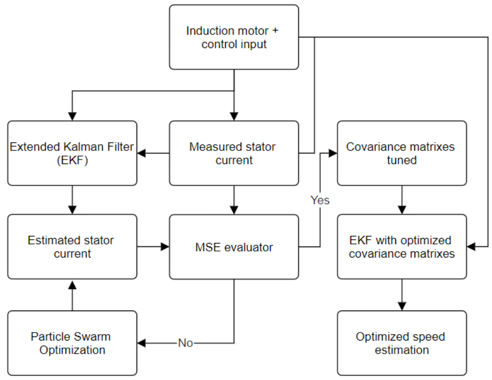

Figure 4. Block diagram of method 2

6. Simulation results

In this section, the induction motor model has been built in MATLAB Simulink as shown in Figure 2. The EKF algorithm was also written in a MATLAB file, moreover the

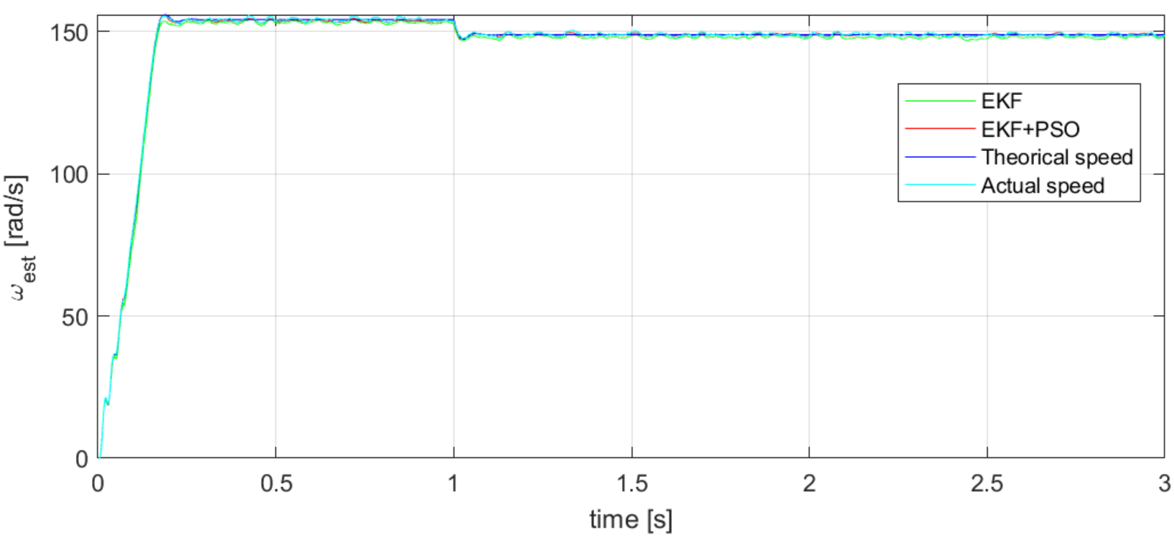

Figure 5. Estimation of the speed with method 1

fitness function which incorporates the EKF algorithm is written as a function file.

Band-limited white noise blocks which generate normally distributed random numbers have been added to the simulation in order to simulate noise and show the effectiveness of the EKF-PSO system. The induction motor has been fed a three-phase sinusoidal voltage input with a sample time of $1 \cdot 10^{-4}$ seconds.

The key problem of the EKF has always been determining the appropriate covariance matrix, since a too high or too low value of covariance matrix can lead to different problems which will constraint the performance of the EKF. For example, if the system noise covariance matrix Q is too low, it may causes lag error, while a too high Q permits the EKF to follow the measurements but produces noisy estimations. Furthermore, when the measurement noise covariance matrix R is too small, it will lead to an unsteady and erratic estimation, and when R is too big, it would lead to lag error, similarly to a too small value of Q. A classical method to determine the value of Q and R is to manually input different values for the covariance matrix but this method can be highly consuming and not very evident for a non-expert user.

Table 1. Manuel inputs of Q and R values

Case	Values of Q and R
1	$q_i = q_\psi = 1 \cdot 10^{-4}, q_\omega = r_i = 1 \cdot 10^{-1}$
2	$q_i = q_\psi = 1 \cdot 10^{-4}, q_\omega = 1 \cdot 10^{-2}, r_i = 1 \cdot 10^{-1}$
3	$q_i = 1 \cdot 10^{-3}, q_\psi = q_\omega = 1 \cdot 10^{-2}, r_i = 1$
4	$q_i = q_\psi = 1 \cdot 10^{-6}, q_\omega = 1 \cdot 10^{-2}, r_i = 1 \cdot 10^{-2}$
5	$q_i = 1 \cdot 10^{-9}, q_\psi = 1 \cdot 10^{-11}, q_\omega = 1 \cdot 10^{-5}, r_i = 1 \cdot 10^{-4}$
6	$q_i = q_\psi = 1 \cdot 10^{-7}, q_w = 1 \cdot 10^{-2}, r_i = 1 \cdot 10^{-1}$
7	$q_i = 1 \cdot 10^{-3}, q_\psi = 1 \cdot 10^{-4}, q_\omega = 1 \cdot 10^{-5}, r_i = 10$
8	$q_i = q_w = 1 \cdot 10^{-3}, q_\psi = 1 \cdot 10^{-4}, r_i = 10$
9	$q_i = 1 \cdot 10^{-3}, q_\psi = 1 \cdot 10^{-4}, q_\omega = 1 \cdot 10^{-2}, r_i = 10$

The MSE of each case shown in is presented in Figure 6. The MSE is calculated by comparing the value of the measured speed and the estimated speed through EKF. The worst case in scenario is case 1 where the MSE is 53.82 while the best case in scenario is case 9 where the MSE is 1.30. Case 1 till 9 have been put in order of their respective MSE however, as seen in Figure 7, the values for Q and R do not exhibit an obvious pattern. This renders the process of tuning the covariance matrices relatively unintuitive towards newcomers. Moreover, the process can be extremely time-consuming. The necessity for a method to determine the values of Q and R is therefore primordial to ensure an accurate estimation. The first method mentioned in section 5 while can only be operated offline where the speed of the rotor can be measured, is an good way to show the effectiveness of the PSO-EKF framework. The PSO will provide us with a more tuned value of Q and R in less time.

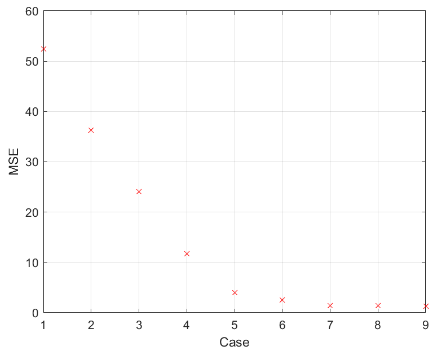

Figure 6. MSE values according to each case

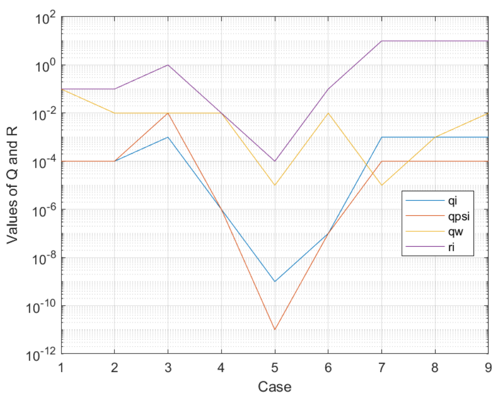

Figure 7. Values of Q and R for each case

Upon running the PSO algorithm, and have been tuned in order to optimized to MSE between the EKF estimation and the actual speed. After 7 iterations, a MSE of 0.17 has been achieved. Figure 8 shows the evolution of the MSE after each iteration. Due to the exponential variation of the inertia weight, the exploration of the swarm was extensive initially, moving forward, the inertia weight experienced a rapid decrease which promotes a faster convergence. The optimized value for and are shown to be $q_i = 4.8 \cdot 10^{-3}$, $q_\psi = 7.1 \cdot 10^{-5}$, $q_\omega = 2.7 \cdot 10^{-3}$, $r_i = 39.4$

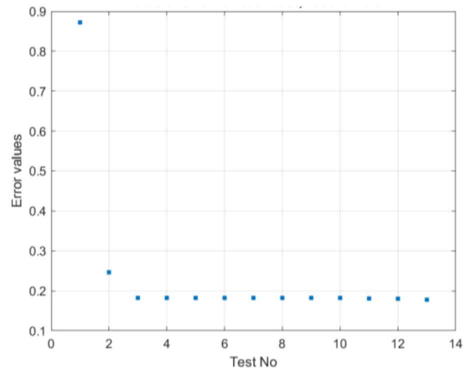

Figure 8. Values for each PSO iterations

As seen in Figure 5, the difference between the green line (EKF without PSO) and the cyan line (actual speed) shows a clear discrepancy while the red line (EKF+PSO) tracks the cyan line while being much smoother than the latter. error between the EKF model and the actual is clearly higher than the error between the EKF+PSO model and the actual speed. Figure 9 reiterates this point.

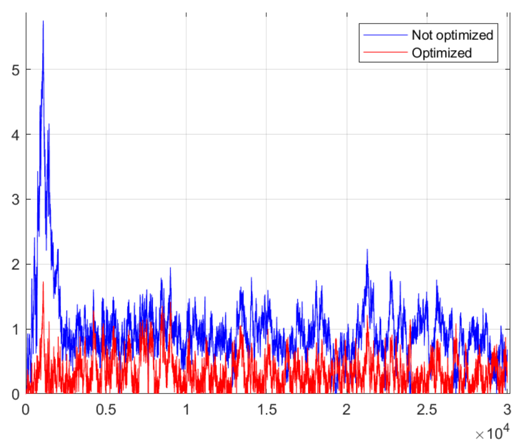

Figure 9. Graph of the error for each model

The sensor less method mentioned in section 5, unlike the first method mentioned above, can theoretically be applied to our problem: estimation of rotor speed without a sensor. Instead of comparing the EKF's rotor speed estimation with the actual rotor speed which we would need a sensor on the motor shaft to retrieve, we would compare the already measured stator current and the EKF's estimation of the stator current. As before, the PSO will algorithmically determine the optimized value for and in order to reduce the optimized value of the MSE between the measured stator current and the estimated stator current. After the PSO has been ran, the following values of Q and R: $q_i = 8.4 \cdot 10^{-3}$, $q_\psi = 1 \cdot 10^{-4}$, $q_\omega = 4.0 \cdot 10^{-3}$, $r_i = 38.2$ have been achieved. The MSE between the actual stator current and the estimated stator current is 0.46 while the MSE obtained between the actual rotor speed and the estimated rotor speed is 0.23. It is noted that the lower the MSE between the stator current and its estimation does not necessarily induce a better estimation of the rotor speed. Another important aspect is the smoothness of the estimation as seen in Figure 10.

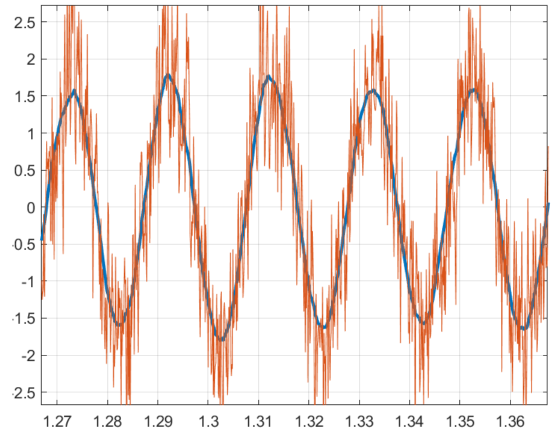

Figure 10. Estimation of the stator current (zoomed in)

A more "accurate" estimation of stator current with lower MSE but higher variation from step to step does not lead to an accurate estimation of the speed. In fact, it creates much higher degree of error in the estimated rotor speed.

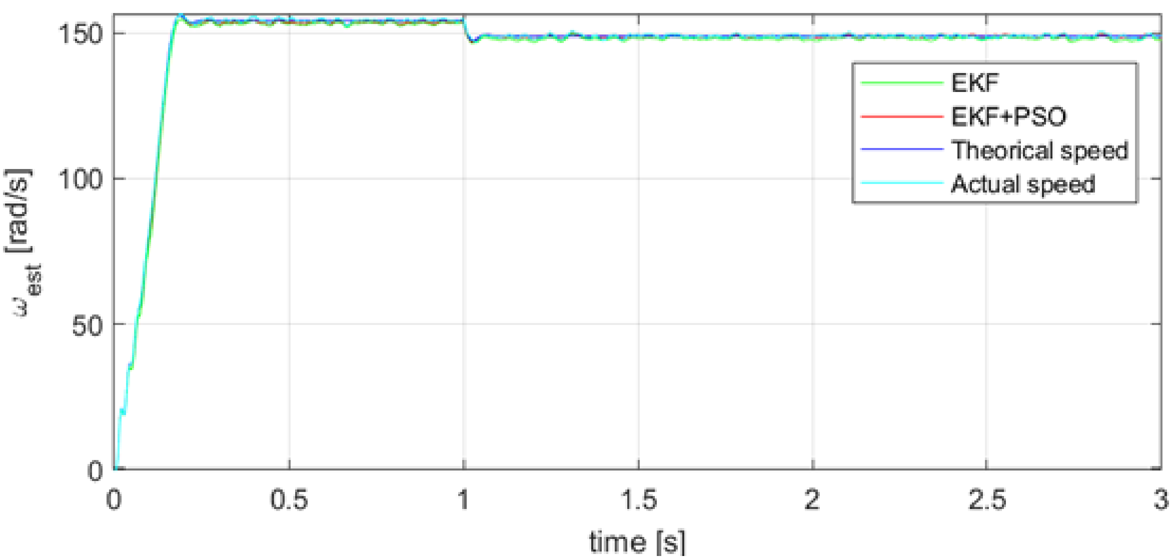

Figure 11. Estimation of speed with the sensor less method

By using the PSO algorithm, a lower MSE between the stator current and estimation can be achieved: 0.389 which gives us better tracking of the stator current as shown in Figure 12. It may follow the measured stator current in closer fashion but does not lead to a more accurate estimated speed as shown in Figure 13.

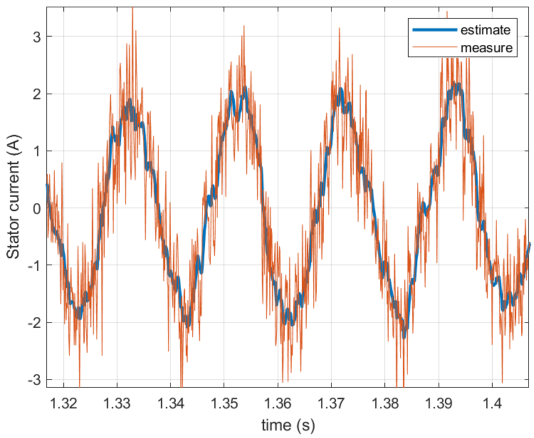

Figure 12. Estimation of stator current (0.389 MSE)

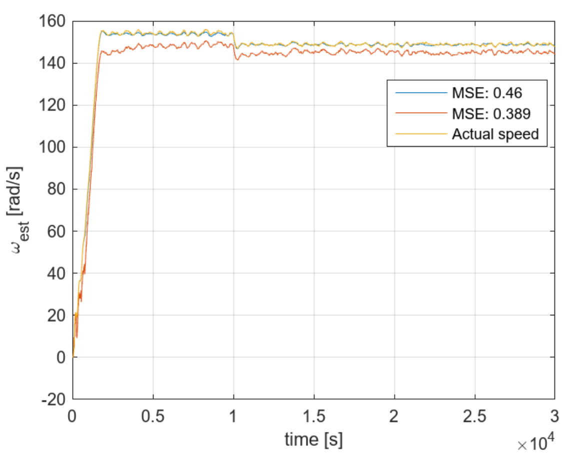

Figure 13. Estimation of speed according to the MSE of stator current estimation

Figure 13 emphasizes the fact that not only the MSE of the estimation of the stator current plays a role in the speed estimation process. Another important factor is the smoothness of the estimation.

In order to use the smoothness of the estimation as a factor in the tuning process, the fitness function must therefore be changed to take into account the smoothness of the curve. The fitness function will then be the following:

$$Fitness = \alpha \cdot MSE\left(i_{estimated} \& i_{measured}\right) + \beta \cdot Var\left(i_{estimated}\right)$$

Whereas,

$$Var\left(i_{estimated}\right) = \frac{1}{N-1}\sum_{i=1}^{N}\left|i_{estimated,i} - \mu_{i_{estimated}}\right|^2$$

With, α and β being weighted coefficients. This is extremely important, as mentioned before, a too small MSE lead to a noisy estimation of the stator current so a weighted coefficient will emphasize the importance of the smoothness of the estimation. After trial and error, α=1 & β=2 have been chosen. Unlike, the fitness where the smoothness of the estimation was not taken into account, the position bound of the PSO had to be extremely narrow, the position bound using a weighted coefficient where the smoothness of the estimation is taken into account can be widen, for example the search space for q_ψ initially set stricter than in method 1 (from 0 till 0.0001) can now be widen (from 0 till 0.1).

To determine the desired criteria, the reference will be the MSE of 0.46 and a variance of $i_{estimated}$ of 1.93. So, the fitness function should be smaller than 4.32. The chosen criteria is then selected to be 4.3.

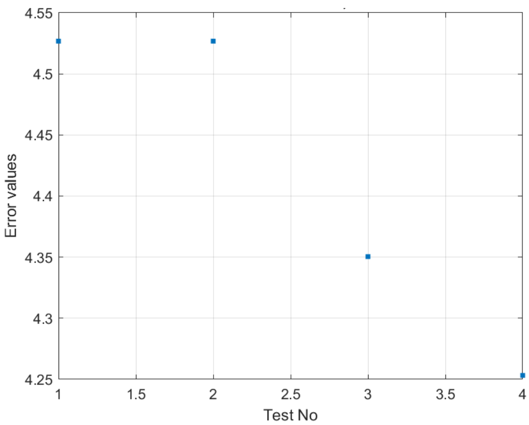

Figure 14. Error values for PSO

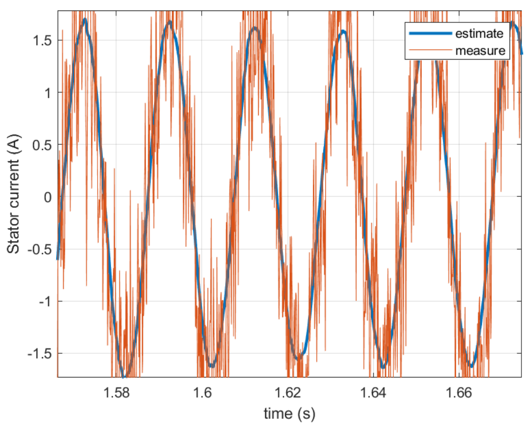

Figure 15. Zoomed in estimation of stator current (0.47 MSE)

As shown in Figure 14, after 4 iterations of the PSO, a fitness value of 4.25 has been achieved. The MSE between the estimated current and measured current is 0.47 higher than the MSE achieved with the prior fitness function. To compensate for this fact, the estimation is then smoother as seen in Figure 15.

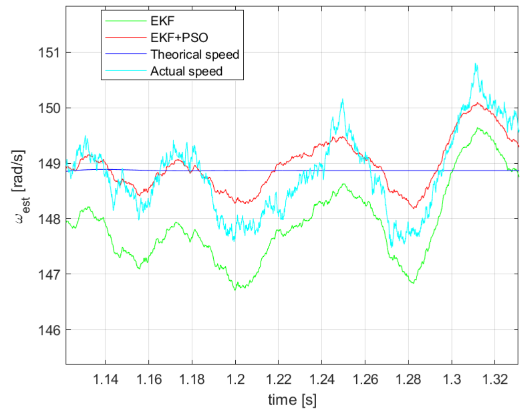

Figure 16: Zoomed in estimation of speed with sensor less method (Smoothness taken into account)

The following covariance matrices are obtained: $q_i = 4.9 \cdot 10^{-3}$, $q_\psi = 6.2 \cdot 10^{-5}$, $q_\omega = 3.8 \cdot 10^{-3}$, $r_i = 39.0$. The MSE between the measured speed and the estimated speed is 0.2086 which is even smaller than when the smoothness of the stator current estimation wasn't taken into account. As depicted in Figure 16, the estimated speed still tracks the actual speed while reducing the noise of the actual speed.

In summary, with both methods, the PSO algorithm have significantly increased the accuracy of the speed estimation. While the sensor less method couldn't reach the same accuracy as the first method, it nevertheless remains a good option for speed estimation without a need for a speed sensor. The estimation of the speed through PSO still tracks the speed whilst reducing its noise as seen in Figure 5, Figure 11 and Figure 16 for the first method and the sensor less method respectively.

7. Conclusion

This paper advocates for the importance of speed estimation in induction motors. An accurate speed estimation is extremely important when using induction motors. The Extended Kalman Filter is primordial in order to estimate the speed of the rotor and to filter out the environment's noise. Along with the PSO algorithm which optimizes the and covariance matrix, which is essential to the operation of the EKF, an accurate speed estimation of the rotor has been achieved. Simulations in MATLAB Simulink have proven that the PSO algorithm can significantly improve upon the EKF by optimizing and . Furthermore, two methods of implementing the PSO into the EKF have been demonstrated. On one hand, in the case where the actual speed is measured, the PSO+EKF framework can efficiently and accurately estimate the speed and reduce its noise. While, on the other hand, in the case of the sensorless induction motor, the PSO+EKF can still estimate the speed accurately, albeit with a lesser degree of accuracy than the former.

References

[1]. Z.M.S. Elbarbary, O.K. Al-Harbi, Saad F. Al-Gahtani, Shaik M. Irshad, Almoataz Y. Abdelaziz, Mahmoud A. Mossa, Review of speed estimation algorithms for three- phase induction motor, MethodsX, Volume 12, 2024.

[2]. C.M.F.S. Reza, Md. Didarul Islam, Saad Mekhilef, A review of reliable and energy efficient direct torque-controlled induction motor drives, Renewable and Sustainable Energy Reviews, Volume 37, 2014, Pages 919-932.

[3]. Shuting Fan, Dawei Meng, Mengmeng Ai, Efficiency analytical of five-phase induction motors with different stator connections for fracturing pump drives, Energy Reports, Volume 8, 2022, Pages 405-413.

[4]. K. Karthikumar, Ahmad Omar Deab, S. Ramesh, A hybrid methodology-based battery power management and torque control of open-end winding induction motor drive for electric vehicles, Journal of Energy Storage, Volume 97, Part A, 2024.

[5]. Tewodros Gera Workineh, Yosef Berhan Jember, Abrham Tadesse Kassie, Evaluation of intelligent PPI controller for the performance enhancement of speed control of induction motor, Scientific African, Volume 22, 2023.

[6]. Erik Etien, Modeling and simulation of soft sensor design for real-time speed estimation, measurement and control of induction motor, ISA Transactions, Volume 52, Issue 3, 2013, Pages 358-364.

[7]. Pavel Karlovský, Jiří Lettl, Application of MRAS Algorithm to Replace the Speed Sensor in Induction Motor Drive System, Procedia Engineering, Volume 192, 2017, Pages 421-426.

[8]. Quanbo GE, Zihao SONG, Bingtao ZHU, Bingjun ZHANG, Adaptive nonlinear Kalman filters based on credibility theory with noise correlation, Chinese Journal of Aeronautics, Volume 37, Issue 6, 2024, Pages 232-243.

[9]. Xuehua Zhao, Dejun Mu, Jiahui Yang, Jiahao Zhang, Rational-quadratic kernel-based maximum correntropy Kalman filter for the non-Gaussian noises, Journal of the Franklin Institute, 2024.

[10]. Becker, A. Kalman Filter from de ground up, 2023, Page 221.

[11]. Brown, R. G., & Hwang, P. Y. C. Introduction to Random Signals and Applied Kalman Filtering with Matlab Exercises. John Wiley & Sons, 2012, Pages 257-260.

[12]. J.R. Perán, L.J de Miguel, Antonio Mendoza, Roberto Arnanz, Alicia Corrales, OVERCOMING SENSOR FAULTS IN CONTROLLED INDUCTION MOTORS USING EKF, IFAC Proceedings Volumes, Volume 38, Issue 1, 2005, Pages 275-280.

[13]. Peter Matisko, Vladimír Havlena, Noise covariances estimation for Kalman filter tuning, IFAC Proceedings Volumes, Volume 43, Issue 10, 2010, Pages 31-36.

[14]. Dragan Ćetenović, Aleksandar Ranković, Junbo Zhao, Zhaoyang Jin, Jianzhong Wu, Vladimir Terzija, An adaptive method for tuning process noise covariance matrix in EKF-based three-phase distribution system state estimation, International Journal of Electrical Power & Energy Systems, Volume 132, 2021.

[15]. Vinay A. Bavdekar, Anjali P. Deshpande, Sachin C. Patwardhan, Identification of process and measurement noise covariance for state and parameter estimation using extended Kalman filter, Journal of Process Control, Volume 21, Issue 4, 2011, Pages 585-601.

[16]. Hadi Nobahari, Alireza Sharifi, A hybridization of extended Kalman filter and Ant Colony Optimization for state estimation of nonlinear systems, Applied Soft Computing, Volume 74, 2019, Pages 411-423.

[17]. K. L. Shi, T. F. Chan, Y. K. Wong and S. L. Ho, "Speed estimation of an induction motor drive using an optimized extended Kalman filter," in IEEE Transactions on Industrial Electronics, vol. 49, no. 1, pp. 124-133, Feb. 2002.

[18]. Anula Khare, Saroj Rangnekar, A review of particle swarm optimization and its applications in Solar Photovoltaic system, Applied Soft Computing, Volume 13, Issue 5, 2013, Pages 2997-3006.

[19]. Aditya Anshul, Anirban Sengupta, PSO based exploration of multi-phase encryption based secured image processing filter hardware IP core datapath during high level synthesis, Expert Systems with Applications, Volume 223, 2023.

[20]. Pei-wen Zhang, Lian-zheng Zhao, Yu Wang, Rui Ding, Fu-min Du, Air route link prediction based on the PSO-CLP model, Journal of Air Transport Management, Volume 120, 2024.

[21]. Renato Barbieri, Nilson Barbieri, Key Fonseca de Lima, Some applications of the PSO for optimization of acoustic filters, Applied Acoustics, Volume 89, 2015, Pages 62-70.

[22]. S. Allaoui, K. Chafaa, Y. Laamari and B. Athamena, "Induction motor state estimation using tuned Extended Kalman Filter," 2015 4th International Conference on Electrical Engineering (ICEE), Boumerdes, Algeria, 2015, pp. 1-5.

[23]. Laamari, Y., Chafaa, K. & Athamena, B. Particle swarm optimization of an extended Kalman filter for speed and rotor flux estimation of an induction motor drive. Electr Eng 97, 2015, pp. 129–138.

[24]. Nooshin Bigdeli, Optimal management of hybrid PV/fuel cell/battery power system: A comparison of optimal hybrid approaches, Renewable and Sustainable Energy Reviews, Volume 42, 2015, Pages 377-393.

Biography :

Dao Dinh Thien , Third year, undergraduate at the Hanoi University of Science and Technology (USTH), majoring in Aeronautical Engineering. Graduated Lycée Français Alexandre Yersin (LFAY) in 2019.

His current research is centered on nonlinear control systems.