DESIGN OF STATE FEEDBACK CONTROLLER WITH OPTIMAL PARAMETERS USING BAT ALGORITHM FOR REACTION WHEEL PENDULUM

Chiem Nguyen Xuan
Department of Automation and Computing Techniques
Le Quy Don Technical University
Hanoi, Vietnam
nguyenxuanchiem83@gmail.com

Thang Le Tran
Controls, Automation in Production and Improvement of Technology Institute
Hanoi, Vietnam
ltranthang@gmail.com

***Abstract* — This paper presents a method of designing a positional stability control system for the Reaction Wheel Pendulum. The Reaction Wheel Pendulum is a single-input, multiple-output (SIMO) nonlinear system. The state feedback controller is designed for this system by choosing the distribution of the solutions of the above characteristic polynomial in the solution space. This original distribution is optimized by the BAT algorithm through the ITAE objective function. The results of the synthetic control law are proven through simulation results and comparison with other control laws. The effectiveness of the rule with optimized parameters is also shown in the results on the embedded control system.**

Keywords* — *Reaction Wheel Pendulum, LQR controller, SIMO, BAT algorithm, pole-assignment controller, ITAE.

I. INTRODUCTION

The reaction wheel pendulum (RWP) is one of the nonlinear systems, which is made up of a combination of an reaction wheel pendulum. A balanced gyro system consists of a pendulum rod with one end fastened to a free axis so that the rod can rotate freely along that axis. The other end of the pendulum rod is fastened to a motor, which is attached to a reaction wheel. The control purpose of this system is to use the torque generated by the reaction wheel to stabilize the pendulum in a vertical position. Like other nonlinear systems such as the inverted pendulum [1], ball stick [2], pendubot [3]…, the balance gyroscopic - reaction wheel system [4] is also a highly nonlinear system, which is difficult to control and has a SIMO structure, i.e. the system has one control variable and two joystick variables (the control variable is the torque generated by the movement of the moving reaction wheel, the two joystick variables are the angle pendulum φ and reaction wheel angle θ as shown in Figure 1). The control target to maintain local stability around the equilibrium position has been published in many studies. In [5], a traditional PID controller was introduced that stabilized the system at the equilibrium position. The LQR controller designed for this system in the study [6, 7] gives relatively good results. In order to overcome the external noise and uncertainty of the model parameters, the fuzzy logic approach is presented in [8, 11]. Many nonlinear control laws have been proposed in many studies such as: slip control described in [9] and feedback linearization used in [10]. Although the quality of the system control is very good, the control law is often complex and affected by the uncertainty of the system. In the study [12], a linear state feedback controller was designed. This control law can work well around the operating point; However, the optimal determination of the poles of the characteristic polynomial has not been considered.

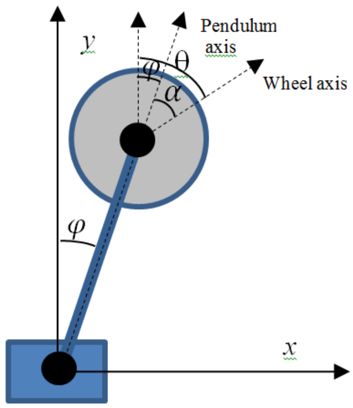

Fig. 1. Model of a Reaction Wheel Pendulum

The controller parameter optimization techniques are presented in the studies [13,14]. But with highly nonlinear system and multivariable objective function, this technique does not give good performance. The optimization algorithm based on Nature-Inspired Metaheuristics is a strong development trend. There have been many optimization algorithms successfully built from the behavior of animals and have been widely published such as: genetic algorithms (GA), ant colony optimization (ACO), bat algorithms (BA), bee algorithms, differential evolution (DE), particle swarm optimization (PSO), harmony search (HS), the firefly algorithm (FA), cuckoo search (CS), and the flower pollination algorithm (FPA), and others [15,16].

This paper presents a method to design a state feedback controller by assigning poles and optimizing its set parameters by using the BAT algorithm based on the ITAE objective function. The controller results are illustrated by simulation on the software and test running the controller on the real system. The efficiency of the optimized control law is shown when compared with some traditional linearized feedback control laws.

II. MATHEMATIC MODEL REVERSE SYSTEM WITH RED WHEEL

The Reaction Wheel Pendulum has two degrees of freedom and the angular coordinates of the pendulum φ

and θ of the reaction wheel ($\theta = \varphi + \alpha$) as shown in Figure 1.

System parameters: $m = m_p + m_r$; $l = (m_p l_p + m_r l_r) / m$;

$J = J_p + m_p l_p^2 + m_r l_r^2$.

Where: m_r= 0.3 - Weight of reaction wheel (kg); m_p=1.0 - Weight of pendulum (*kg*); l_p=0.28 - Distance from the shaft to the center of mass of the pendulum (*m*); l_r - 0.3 - Distance from the axis of rotation to the center of mass of the reaction wheel (*m*); l - Distance from the shaft to the center of mass of the pendulum and rotor (*m*); J_p - Moment of inertia of the pendulum (*kg/m*^{2}); J_r - Moment of inertia of the wheel (*kg/m*^{2}); θ - reaction wheel deflection angle (*rad*); φ - angle of deflection of the pendulum (*rad*); g - Acceleration due to gravity (*m/s*^{2});

m_p=1.0(*kg*); m_r=0.3(*kg*); l_p=0.28 (*m*); l_r= 0.3 (*m*); J_p= 0.28 (*m*); J_r= 0.28 (*m*) and g= 9.8 (*m/s*^{2})

The kinetic energy T of the system is the sum of the kinetic energy of the pendulum and the kinetic energy of the reaction wheel and can be written as the above quantities:

$$T = \frac{1}{2} J \dot{\varphi} \quad \frac{1}{2} \tag{1}$$

The potential energy V of the system is only due to gravity, ignoring the elasticity of the pendulum:

$$V = mgl \cos\varphi \tag{2}$$

It is clear that the potential energy of the system does not depend on the reaction wheel position because the mass of the reaction wheel is distributed symmetrically about its axis of rotation. Using the Lagrange equation [7] for this system, we have the following Lagrange function of the system:

$$L = T - V = \frac{1}{2} J \dot{\varphi} \quad \frac{1}{2} \quad \ldots \cos\varphi \tag{3}$$

Taking the partial derivative of the Lagrange function for each variable, we have:

$$\frac{\partial L}{\partial \dot{\varphi}} = J \dot{\varphi} \quad \frac{\partial L}{\partial \varphi} \quad gl \sin\varphi; \frac{\partial L}{\partial \dot{\theta}} = \ldots \quad \theta \tag{4}$$

In this case, the torque produced by the motor τ acts on the reaction wheel and - τ acts on the pendulum. Neglecting the friction force and the electromotive force of the DC motor, the torque is given by:

$$\tau = kI \tag{5}$$

where k is the torque constant of the motor and I is the motor amperage. From Equation 4 we get:

$$\begin{cases} \ddot{\varphi} \quad \frac{mgl}{J} \sin\varphi = -\frac{k}{J} I \\ \ddot{\theta} \quad \frac{\quad}{J_r} \end{cases} \tag{6}$$

Convert the system of equations (6) to the system of state equations with $x_1 = \varphi$; $x_2 = \dot{\varphi}$ when linearizing at the equilibrium point, we get equation (7):

$$\ldots u \tag{7}$$

Where $x = \begin{bmatrix} x_1 \\ x_2 \\ x_3 \end{bmatrix}; A = \begin{bmatrix} 0 & 1 & 0 \\ mgl/J & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix}; B = \begin{bmatrix} 0 \\ -k/J \\ k/J \end{bmatrix}$

and $u = I$.

Equation (7) with a control signal as input (current I), two signals as pendulum deflection and reaction wheel rotation as output, represents a SIMO nonlinear system with one input and two outputs.

III. BASICS OF BAT ALGORITHM

The standard bat algorithm was developed by Xin-She Yang [16]. The main characteristics in the BA are based on the echolocation behavior of microbats. As BA uses frequency tuning, it is in fact the first algorithm of its kind in the context of optimization and computational intelligence. Each bat is encoded with a velocity v_i^t and a location x_i^t, at iteration t, in a d - dimensional search or solution space. The location can be considered as a solution vector to a problem of interest. Among the n bats in the population, the current best solution x^* found so far can be archived during the iterative search process.

Based on the original paper by Yang, the mathematical equations for updating the locations x_i^t and velocities v_i^t can be written as:

$$f_i = f_{\min} + \left(f_{\max} - f_{\min}\right)\beta,$$
$$v_i^t = v_i^{t-1} + \left(x_i^{t-1} - x_*\right) f_i,$$
$$x_i^t = x_i^{t-1} + v_i^t,$$

where $\beta \in [0; 1]$ is a random vector drawn from a uniform distribution.

In addition, the loudness and pulse emission rates can be varied during the iterations. For simplicity, we can use the following equations for varying the loudness and pulse emission rates:

$$A_i^{t+1} = \alpha A_i^t \text{ and } r_i^{t+1} = r_i^0 \left[1 - \exp(-\gamma t)\right],$$

where $0 < \alpha < 1$ and $\gamma > 0$ are constants.

The pseudo code of the basic bat algorithm is presented in Algorithm 1. The main parts of the bat algorithm can be summarized as follows:

First step is initialization (lines 1-3). In this step, we initialize the parameters of algorithm, generate and also evaluate the initial population, and then determine the best solution x_{best} in the population.

Algorithm 1 Original BAT algorithm

Input: Bat population x_i=(x_{i1}, …, x_{iD}) for i = 1…Np MAX_FE

Output: The best solution x_{best} and its corresponding value f_{min}= min(f(x)).

1 : init_bat() ;

2 : eval=evaluate_the_new_population ;

3 : f_{min}=find_the_best_solution(x_{best}) ; {initialization}

4 : **while** termination_condition_not_meet **do**

5 : **for** i=0 **to** N_p **do**

6 : y= improve_the_best_solution(x_{best});

7 : **if** rand(0,1)>r_i **then**

8 : y= improve_the_best_solution(x_{best});

9 : **end if** {local search step}

10: f_{new}=evaluate_the_new_solution(y);

11: eval=eval+1 ;

12: **if** $f_{new} \le f_i$ **and** N(0,1)< A_i **then**

13: $x_i = y$; $f_i = f_{new}$;

14: **end if** {save the best solution conditionally}

15: f_{min} = find_the_best_solution (x_{best});

16: **end for**

17: **end while**

The second step is: generate the new solution (line 6). Here, virtual bats are moved in the search space according to updating rules of the bat algorithm.

Third step is a local search step (lines 7-9). The best solution is being improved using random walks.

In forth step evaluate the new solution (line 10), the evaluation of the new solution is carried out.

In fifth step save the best solution conditionally (lines 12-14), conditional archiving of the best solution takes place.

In the last step: find the best solution (line 15), the current best solution is updated.

IV. DESIGN A STATE FEEDBACK CONTROLLER WITH POLE ASSIGNMENT METHOD AND PARAMETERS OPTIMIZATION USING BAT ALGORITHM FOR RWP

The state feedback control structure diagram for RWP system has the form as shown in Figure 2. The design of state feedback control law using pole assignment method with parameter optimization by BAT algorithm consists of two stages: (i) Design state feedback controller by pole assignment method; (ii) Optimizing the controller parameters of the control rule by the BAT algorithm.

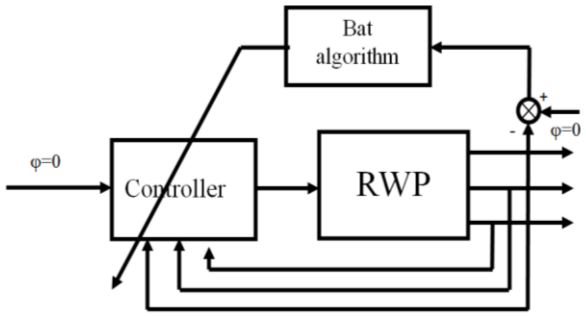

Fig. 2. Diagram of control structure RWP parameter adjustment by BAT algorithm

A. Synthesis of feedback control law with parameter optimization by BAT algorithm.

Consider a control system in the form of a system of state equations as (7). From the block diagram of the system shown in Figure 2 with state feedback control has the following form:

$$u = -Kx \tag{8}$$

where K is a constant feedback gain vector. The input to the control system is assumed to be zero. The purpose of this system is to return all states to zero when the states are perturbed. Then, the closed-loop system according to the control law (8) is

$$K)x \tag{9}$$

The fully closed-loop dynamics are determined by the matrix (A - BK), and the stability of the closed-loop system and the rate of adjustment of x to zero are determined by the eigenvalues of this matrix, are called the poles of the closed-loop system. In particular, system (9) is (asymptotically) stable if and only if all the eigenvalues (s) of the matrix (A-BK) are in the domain Re(s) < 0. So, from the system of equations (9) we use Laplace transform to get the characteristic polynomial of the system of the form $r(s)$ = det(sI-A+BK), we get:

$$r(s) = s^3 + \frac{k(K_3 - K_2)}{J}s^2 - \frac{kK_1 + mgl}{J}s - \frac{mglkK_3}{J^2} \tag{10}$$

For asymptotically stable system (10) we will choose the eigenvalues of (A - BK) for a given pair (A, B). That is, the characteristic polynomial has the form R(s) = (s-λ_1) (s-λ_2) (s-λ_3). With the roots of the characteristic polynomial selected as follows: λ = [-2+j -2+j -3] - where j is an imaginary number. Identifying the parameter with the characteristic polynomial (10) with R(s), we get the matrix K as follows:

$$K = \begin{bmatrix} -0.0865 & -0.0066 & -0.0002 \end{bmatrix} \tag{11}$$

But the choice of eigenvalues or solutions λ of the characteristic polynomial (10) depends on the dynamic properties of the system. Therefore, determining the solution to improve control quality requires a deep understanding of the control object as well as the effects caused by noise.

For the purpose of reducing the transition time of the response of state variables, keeping the static error zero for all states, we propose an objective function of the form that is the integral function of the product of squares of time and the sum of all states absolute error has the form.

$$ITAE: \quad F = \int t^2 \left(|e_1(t)| + |e_2(t)| + |e_3(t)| \right) dt \tag{12}$$

Where $e_i(t) = x_i - 0$ with $i = 1:3$.

The algorithm to optimize controller parameters through three coefficients K_1, K_2, K_3 is implemented by the BAT algorithm. The steps of the algorithm to find the value of the parameter set by the BAT algorithm include the following steps:

1. Initialize the population of bats (n) with the loudness (A) and pulse emission rates (r), the bats have

random position (x_i) and velocity (v_i) for all 3 parameters K_1, K_2, K_3.

2. Calculate the value of the ITAE objective function of all bats in the initial population.

3. Compare the values of the objective function just found to find the bat (x_{best}) for the best objective function value.

4. Update pulse frequency Q and velocity v of all bats according to the following equation:

$$Q_{k+1} = Q_{\min} + (Q_{\min} - Q_{\max}) * rand;$$
$$v_{k+1} = v_k + (x_k - x_{best})Q_{k+1}$$

5. Update the positions of all bats according to the following equation:

$$x_{k+1} = x_k + v_k$$

6. Update the position of the bat if the pulse width (r) is less than the pulse width of the randomly generated signal (rand).

$$\text{if}(rand > r)$$
$$x_{k+1}^{new} = x_{best} + \alpha.rand$$

7. Check the condition, $(rand < A \,\&\, f(x_{k+1}^{new}) < f(x_{k+1}))$, acceptance new population and increase the pulse width (r) and reduce the echo (A).

8. Check if the best value of the new position in the bat population is less than the required value ($f_{\min}$), then the algorithm terminates, otherwise repeat step 4.

To optimize the controller's parameters, the authors choose the optimal solution which is to bring the pendulum from the initial position $X_0 = [0.02\ 0\ 0]^T$ to the equilibrium position $[0\ \ 0\ \ 0]^T$ in *4(s)* such that the objective function value is less than 3 ($f_{\min} = 3$). The initial controller parameter value is selected at (11). Initial data for the BAT algorithm include: Number of bats: 30 (bats), magnitude A = 0.3 and pulse rate $r = 0.3$. After optimizing the controller with the above conditions, we get:

$$K = \begin{bmatrix} -0.6392 & -0.0851 & -0.0366 \end{bmatrix} \quad (13)$$

The solution of the characteristic polynomial of the system is:

$$\lambda = \begin{bmatrix} -6.9236 + 5.6084j & -6.9236 - 5.6084j & -39.1409 \end{bmatrix}.$$

From the found solutions, it is clear that the real parts of the solutions are negative, so the system is asymptotically stable.

B. LQR optimal control law for RWP

From the system of linear equations of state (10), we need to find the matrix K of the optimal control value:

$$u(t) = -KX(t)$$

satisfying quality norm J reaching the minimum value:

$$J = \int_0^\infty (X^T QX + u^T Ru)dt.$$

Where Q is a positive determinant (or semi-positive), R is the positive determinant matrix. The optimal K matrix is defined by the Riccati equation of the form:

$$K = R^{-1}B^T P.$$

The matrix P must satisfy the equation Riccati:

$$PA + A^T P + Q - PBR^{-1}B^T P = 0 \quad (14)$$

The control law with the system parameter as above receives the control law according to the LQR method with matrix values Q and R:

$$Q = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 0.1 & 0 \\ 0 & 0 & 0.1 \end{bmatrix}; \ R = 10;$$

We receive: $u(t) = -2.0843x_1(t) - 0.2543x_2(t) - 0.1x_3(t)$.

V. SIMULATION AND EXPERIMENTAL RESULTS

A. Simulation results

Perform system simulation on Matlab software. Figures show the response of the inverted pendulum deflection angle (Fig. 3), the response of the inverted pendulum angular velocity (Fig. 4) and the reaction wheel pendulum speed (Fig. 5) when the initial value of the system is out of position. equilibrium position 0.02 (rad.) with 3 different control rules: Method-PP- response with feedback control law assigning poles; Method-LQR-responds to the LQR control law and Method-PPBAT responds to the proposed control rule. From the results of the deflection angle response, it is clear that the transient time of the parameter optimized controller (0.526 *s*) is better, compared to the other controller: LQR (0.6 *s*), PPBAT (2.6 *s*). Although the deflection angle response has a higher overshoot (Fig. 3), the angular velocity response (Fig. 4) and the reaction wheel velocity (Fig. 5) have a smaller overshoot.

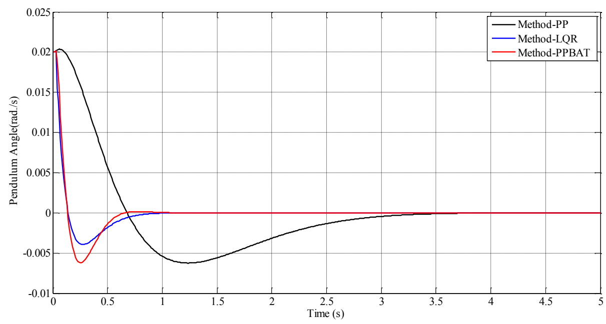

Fig. 3. Inverted pendulum angle response

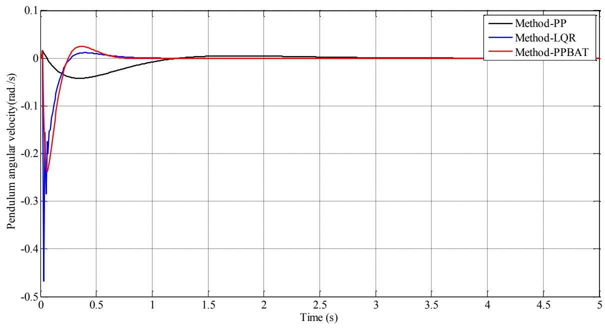

Fig. 4. Angular velocity response of the inverted pendulum

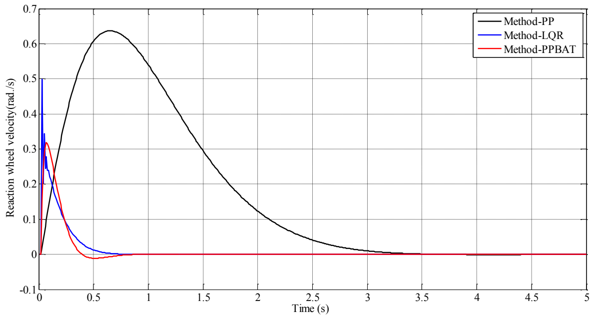

Fig. 5. Reaction wheel pendulum speed response

B. Experimental results

The controller is run in real time on the model made by the author's team (Fig. 6), Arduino Mega 2560 Embedded Board, H7A bridge circuit, the sensor is two incremental encoders 600 pulse and 24V, 25W, 3000rpm DC motor.

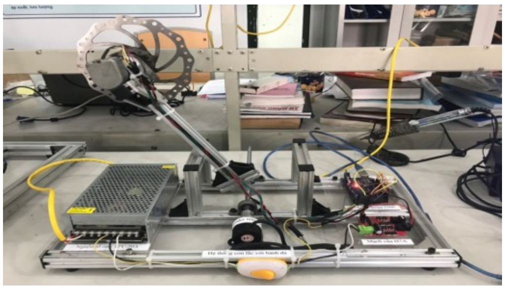

Fig. 6. Experimental model of the reaction wheel pendulum

To monitor, calibrate the system and store embedded data, the author has also designed an interface to connect to the embedded system through RS232 serial communication using Lab view software. The software interface is shown in Figure 8.

The results of running the experimental model after building an embedded control system using the proposed controller are shown in Figure 7, 8:

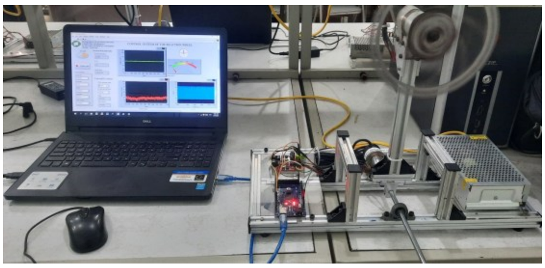

Fig. 7. Stable experimental model when using the proposed controller

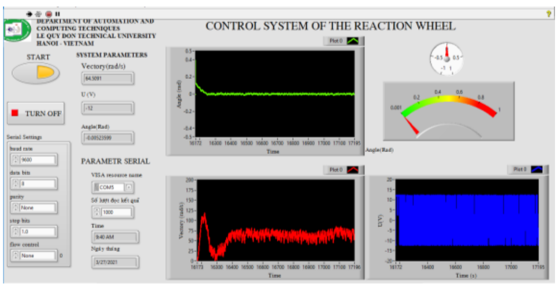

Fig. 8. Results on the computer interface of the experimental model

Figure 8 shows the results when stabilizing the pendulum when deviating from the working point by an angle of 0.1 (*rad*). From the results, we see that the pendulum is stable at the desired position, but the reaction wheel is still spinning due to the relative encoder, so determining the vertical position is not accurate.

CONCLUSION

In the study, synthetic state feedback controller controllers with parameter optimization for RWP were presented. By using the BAT algorithm with the proposed objective function, the authors found a better set of parameters with transient time and systematic static error than parameter selection by assigning parameters. pole. At the same time, the results compared with the LQR controller give better results in terms of transient time and amplitude of oscillation. To demonstrate the efficiency of the controller is synthesized. The authors have designed the above control rule for an embedded RWP system. The test results show the controller's performance on the integrated embedded system. The error responds well to the requirements of the real system. In the next studies, the authors will propose solutions to build a fast-acting nonlinear controller to improve the control system quality for the magnetic elevator system and improve the embedded control system for better performance.

REFERENCES

[1] Deep Ray, Ritesh Kumar, Praveen. C, Mythily Ramaswamy, J.-P. Raymond, Linear conreil of inverted pendulum, IFCAM Summer School on Numerics and Control of PDE, 2013.

[2] Zulhisyam Salleh, Ahmad Nasharuddin A. Rashid, Fizatul A. Patakor, Ball And Beam Educational Tool of Advanced Control System Laboratory, December 2013.

[3] Emmanouil Kourtikakis, Emmanouil Kapellakis, John Fasoulas, and Michael Sfakiotakis, An Embedded Controller for the Pendubot, September 2016.

[4] Karl J. Astrom, Daniel J. Block, Mark W.Spong, The Reaction Wheel Pendulum, Morgan & Claypool Publishers, Synthesus Lectures on Controls and Mechatronics, 2010.

[5] Hyun Woo Kim, Jae Won An, Hang Dong Yoo, Jang Myung Lee, Balancing Control of Bicycle Robot Using PID Control, ICCAS, 2013.

[6] Kiattisin Kanjanawanishkul, LQR and MPC controller design and comparison for a stationary self-balancing bicycle robot with a reaction wheel, Kybernetika, Vol. 51, No.1, 173-191, 2015.

[7] Adam Owczarkowski; Marcin Lis; Piotr Kozierski. Tracking control of an inertial wheel pendulum byLQR regulation. 2014 19th International Conference on Methods and Models in Automation and Robotics (MMAR)

[8] Víctor Daniel Correa-Ramírez, Didier GiraldoBuitrago y Andrés Escobar-Mejía, Fuzzy control of an inverted pendulum Driven by a reaction wheel using a trajectory tracking scheme, TecnoLógicas, ISSN 0123-7799 - ISSN-e 2256-5337, Vol. 20, No. 39, mayo - agosto de 2017.

[9] Iriarte R, Aguilar LT, Fridman L. Second-order sliding-mode tracking controller for inertia wheel pendulum. J Franklin Inst. Elsevier. 2013;350: 92–106

[10] Enev S. Feedback linearization control of the inertia wheel pendulum. Cybernetics and Information Technologies. De Gruyter. 2014;14(3): 96–100.

[11] Scott C. Brown, Kevin M. Passino, Intelligent Control for an Acrobot, Journal of Intelligent and Robotic System, Vol. 18, pp. 209-248, 1997.

[12] Montpellier, France,Ahmed Chemori and Sébastien Krut. Control of the Underactuated Inertia Wheel Inverted Pendulum for Stable Limit Cycle Generation. Advanced Robotics Volume 23, 2009 - Issue 15.

[13] G. Rigatos, M. Abbaszadeh and M.A Hamida. Nonlinear optimal control for the inertia wheel inverted pendulum. CYBER-PHYSICAL SYSTEMS https://doi.org/10.1080/23335777.2019.1678199.

[14] Toussaint GJ, Basar T, Bullo F. H∞ optimal tracking control techniques for nonlinearunderactuated systems. Proceedings IEEE CDC 2000, 39th IEEE Conference on Decision and Control; 2000 Dec; Sydney Australia.

[15] Xin-She Yang. Bat Algorithm: Literature Review and Applications. Int. J. Bio-Inspired Computation, Vol. 5, No. 3, 2013. page: 141-149

[16] Xin-She Yang. Nature-Inspired Optimization Algorithms. Elsevier First edition 2014. 258 page