

Real Time Estimation of Kalman Filter Parameters using the Genetic Algorithm for Optimum Balancing Controller of Two-Wheel Robotic System

Lina N. Tofiq

Electrical Engineering Dept. /College of Engineering
Salahaddin University
Hawler, Kurdistan Region, Iraq
lina.alqasab@gmail.com

Fadhil T. Aula

Electrical Engineering Dept. /College of Engineering
Salahaddin University
Hawler, Kurdistan Region, Iraq
fadhil.alula@su.edu.krd

Abstract— In this paper, a real time estimation procedure is presented in a self-balancing two-wheel robotic system. Such a robotic system has a potential problem due to its unstable state for controlling the balance while the robot is moving forward and/or backward on two-wheel. The controller includes two subsystems: self-balance, which prevents the system from falling down when it moves, and yaw rotation, which regulates a wheel angle when it turns right and left. The genetic algorithm (GA) is used for estimating and tuning Kalman parameters: measurement noise covariance, R, and process noise covariance, Q. The two-wheel robotic system that has been used in this research is adapted with Inertial Measurement Unit (IMU) sensor. The algorithm is simulated in MATLAB software and the results have shown the effectiveness of GA for getting the optimum tuning of Q and R for the self-balancing robotic system.

Keywords; Genetic algorithm; Kalman filter; optimal controller; robotic system; self-balancing.

I. INTRODUCTION

Nowadays, robots are used in many fields such as in automotive, manufacturing, entertainment, educations, military, etc. Many researches have been underway for developing the advanced robotic systems. These systems require the implementations of different aspects in controller concepts. The interesting of designing advanced robotic systems lead to more complicated controller, hence, more challenges to control engineering.

Over many types of robotic systems, a self-balancing robotic system gained more focus by the researchers as well as manufactures due to its systematic controller behavior. This robotic system has the ability to balance itself while moving on two wheels [1,2].

The self-balancing two-wheel robot requires only two points for contacting with the surface. The main idea of this robot is driving the wheels in the direction in which the robot tilts. Controlling the wheel movements is the key to staying the robot stand up while it movies. This action is close to the inverted pendulum model in control theory [3].

The controller of the self-balancing two-wheel robot is an interested area for researchers. A classical and linear multivariable control techniques [4,5], fuzzy-neural control method [6], nonlinear back stepping control implementation

[7], a PID controller implementation in the self-balancing robot [8,9], and a fuzzy PD control method reported in [10]. And many other methods are available in this field aimed to design better controller performance, faster response, and simplified controller components.

In this paper the optimal controller based on Kalman filter is used. Although, using a Kalman filter was reported in other studies [11,12], but the tuning of the filter parameters subjected to estimations and/or assumptions. The tuning of Kalman parameters is the main challenge of designing the optimum filter. This paper focuses on tuning the Kalman filter using the genetic algorithm [13,14] for optimum controller of the self-balancing two-wheel robotic system.

The paper is organized as follows: the general system model of two-wheel balance robot is presented in Section II. The mathematical model is given in Section III. The general concepts of the Kalman filter and the genetic algorithm are presented in Section IV and V, respectively. Implementation, simulation, and results are presented in Section VI. Finally, the conclusions are derived in Section VII.

II. A TWO-WHEEL SELF-BALANCED ROBOTIC SYSTEM

The structure of the two-wheel self-balanced control consists of three main subsystems: motors, sensors, and the controller. The simplify block diagram of such a robotic system is shown in Fig. 1.

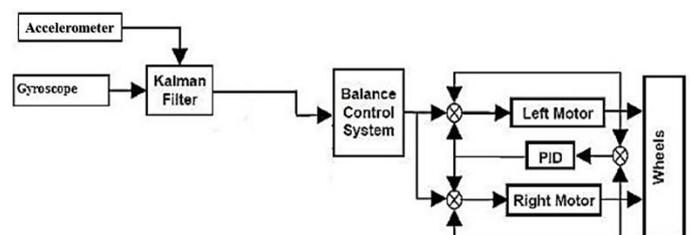


Figure 1. The two-wheel self-balanced robotic system block diagram.

The implementation of IMU is to give a gyroscope and accelerometer sensors of the system balance signals, their functions are as follows [15]:

1. Gyroscope: this sensor measures the angular velocity of the two-wheel robot, and then it sends the data to the controller.
2. Accelerometer: this sensor measures the acceleration of the robot. This includes motion and gravitational accelerations.

It should be noticed that, the accelerometer is responsible for providing the tilt angle, angular velocity θ , while the gyroscope acts as the integration of the angular velocity, that is, $\dot{\theta}$.

The motors have encoders; these encoders convert the rotational angular velocity of each motor to the corresponding digital signals.

The controller for both motors and their encoders is a PID controller, which is sufficient to give the required controller signals to the motors.

The self-balancing control highly depends on the measurement of the sensors. For example, to maintain the upright position of the robot, the tilt angle, θ , needs to be corrected by wheel motion. The best controller, therefore, highly depends on the accuracy of the measurement of this angle which is done by gyroscope and accelerometer sensors of IMU. But, due to noise accompanied with these sensors' measurements, therefore, a filter is required. The Kalman filter is implemented here as this filter.

The filter takes the gyroscope and accelerometer measurements, θ and $\dot{\theta}$, as inputs and provides the estimated angle θ_{est} [16].

III. DYNAMICAL MODEL OF THE SELF-BALANCING ROBOTIC SYSTEM

The linearized model of the robotic system in Fig. 1 can be given by the following state-space representation [2,16]

$$\begin{bmatrix} \dot{x} \\ \dot{v} \\ \dot{\theta} \\ \dot{w} \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & \frac{2k_m k_e}{Rr^2\alpha} (M_p l_r - I_p - M_p l^2) & \frac{1}{\alpha} M_p^2 g l^2 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & \frac{2k_m k_e}{Rr^2\alpha} (r\beta - M_p l) & \frac{1}{\alpha} M_p g l \beta & 0 \end{bmatrix} \begin{bmatrix} x \\ v \\ \theta \\ w \end{bmatrix} + \begin{bmatrix} 0 \\ \frac{2k_m}{Rr\alpha} (I_p + M_p l^2 - M_p l r) \\ 0 \\ 2k_m (M_p l - r\beta) \end{bmatrix} u(t) \quad (1)$$

where x and v are the horizontal displacement and velocity of the center of the gravity of the robot. θ is the tilt angle, w is the angular velocity, $u(t)$ is the motor input voltage. M_p is the mass of the robot, g is the gravity, I_p is the inertia of the robot's body, l is the distance between the center of the wheel and the center of the robotic system gravity, k_m is the motor torque constant, k_e is the back *emf* constant, R is the terminal resistance and r is the radius of the wheel. and

$$\alpha = I_p \beta + 2M_p l^2 \left(M_w + \frac{I_w}{r^2} \right) \quad (2)$$

$$\beta = 2M_w + \frac{2I_w}{r^2} + M_p \quad (3)$$

where M_w and I_w is the mass and inertial of the wheel, respectively.

IV. KALMAN FILTER

A continuous time-invariant linear system is given by

$$\begin{aligned} \dot{x}(t) &= Ax(t) + Bu(t) \\ y(t) &= Cx(t) + Du(t) \end{aligned} \quad (4)$$

where $u(t)$ is the input, $x(t)$ is the state vector, $y(t)$ is the output vector, A is the system matrix, B is the input matrix, C is the measurement matrix, and D is the feedforward matrix.

Fig. 2 shows the block diagram of the Kalman filter in MATLAB simulation [17,18]. Process noise is added to the control input u ; this distorted signal is then sent to the plant. Noise is likewise added to the true output of the plant, y , to represent noisy measurement y_v . Both signals u and y_v are considered as inputs to the Kalman filter, which outputs the estimated output y_e based on the recursive least-squared filter. The Kalman filter combines both a plant model and measurements to make an optimal state estimation.

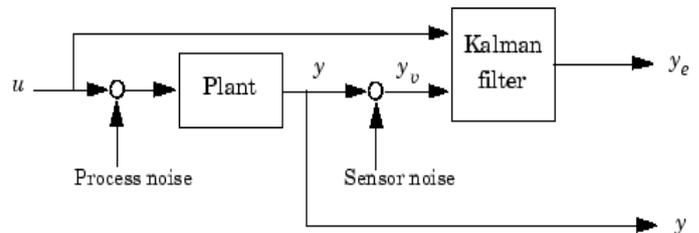


Figure 2. Block diagram of Kalman filter

The process noise is a Gaussian noise, w , representing the system disturbance, and the sensor (or measurement) noise, v , represents the model inaccuracies. Both w and v have a mean value of zero and the following covariance matrices:

$$\begin{aligned} E[w \cdot w^T] &= Q \\ E[v \cdot v^T] &= R \end{aligned} \quad (5)$$

where E denotes the expectation operator and T means the matrix transpose. In this paper, the values of Q and R are estimated using genetic algorithm.

V. GENETIC ALGORITHM

The genetic algorithm (GA) is based on Darwin's evolution theory [18]. GA is an optimization algorithm based on the mechanics of natural genetic and natural selection. GA is the ideal for optimization problem on hand due to their following characteristics:

1. GA is very efficient when large space is involved
2. GA does not require the computation of objective function gradients
3. GA works well with stochastic objective functions
4. The optimization computation can easily be parallelized by GA

Generally, GA performs as follows. An initial population of individual is randomly chosen in the search space, thus composing the initial generation. The population evolves by repeatedly applying three operators on the population, therefore, propagating it through the generations. These operators according to their order of execution are [19]:

1. Selection. Individuals are copied into the next generation with probability relative to their fitness. The higher fitness has the best chance for survival of the next generation in the process. Thus, ‘bad’ individuals are more likely to ‘extinct,’ and ‘good’ individuals will have more ‘descendants’ in the next generation.
2. Crossover. The crossover increases and maintains the diversity of the entire population over the entire run. This is due to fact that a population with higher diversity has the ability to explore a wider range of search space.
3. Mutation. A bit is randomly selected within the chromosome string and mutated. Mutation is typically performed at a very low rate.

VI. SIMULATION AND RESULTS

The system model described in (1), the Kalman filter represented in Fig. 2, and the GA are implemented in the MATLAB software, control toolbox, and MATLAB genetic algorithm function [20]. The values of the parameters in the dynamical modeling of the two-wheel balancing robotic system are given in Table I.

TABLE I. VALUE OF PARAMETERS

Element	Value	Element	Value
M_p	1.13	I_p	0.0041
l	0.07	M_w	0.03
K_m	0.006123	K_e	0.006087
R	3	r	0.051
g	9.81	I_w	0.000039

The values of the Q and R initially set to 1, and the input signal, for the simulation purpose, is chosen to be a sinusoidal signal. The randomness signals that are chosen to be equivalent to the process noise, w , and the sensor noise, v , and consequently to the measurements of IMU gyroscope and accelerometer sensors are Gaussian random generation multiplication by the values of Q and R , respectively.

The MATLAB simulation has been carried out using the above mentioned values and MATLAB genetic algorithm function “gaoptimset.” The main “gaoptimset” parameters set as follows: crossover fraction criteria set at 0.48, and generated for 50 iterations. These two values were set from trial and error method, which gave the best results in optimization of the genetic algorithm. From this function, the simulation was run for finding the optimal angle θ and Kalman filter estimator θ_{est} . The performance of the GA is shown in Fig. 3, and Fig. 4 shows the results of the output of the system with the use of the Kalman filter.

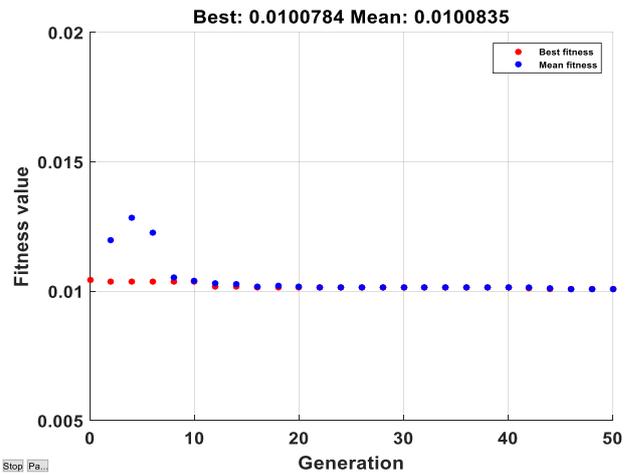


Figure 3. The convergence characteristics of GA.

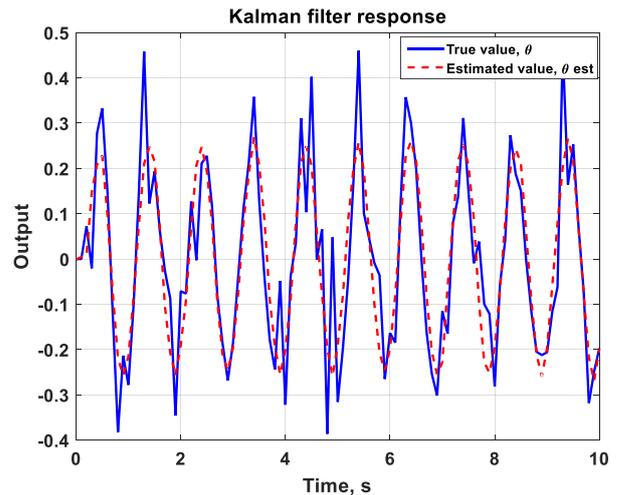


Figure 4. The Kalman filter optimization between true value of θ and estimated value of θ_{est} .

The comparison between the estimation error covariance is shown in Fig. 5.

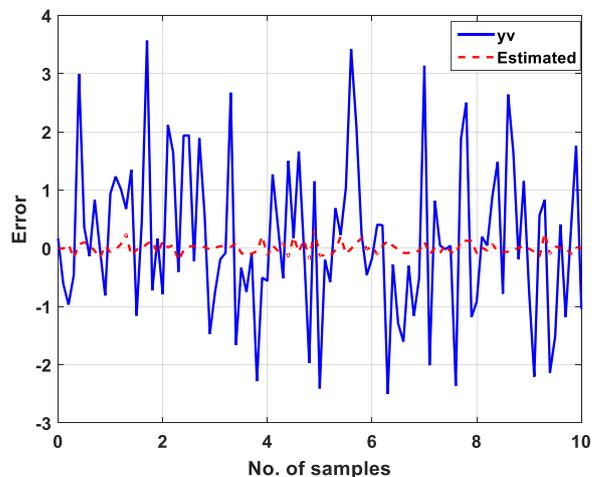


Figure 5. The error covariance of the Kalman Filter

The simulation results show the optimization of the optimum values of Q and R , and the estimation error covariance are shown in Table II.

TABLE II. OPTIMAZATION RESULTS

Output	Q	R	Estimation Error Covariance
Initial	1	1	0.0482
Tunes	0.2079	1.8687	0.0101

VII. CONCLUSION

In this paper, the genetic algorithm has been studied for optimizing the values of Q and R of the Kalman filter. The Kalman filter is used for reducing the effects of uncertainty of both sensor readings of IMU of the two-wheel balanced robotic system. The optimization of the controller is essential for making the robot move in any direction without falling down. The results from the MATLAB simulation showed that the GA can be used as an estimator to the variation of the stochastic behavior of the sensor readings. Also, it was noticed by increasing the crossover criteria of the GA, the better estimation may be achieved, and this is left for further studying in the future.

REFERENCES

- [1] F. Grasser, A. D'Arrigo, S. Colombi, and A. C. Rufer, "JOE: A mobile, inverted pendulum," *IEEE Transactions on Industrial Electronics*, vol. 49, no. 1, pp. 107–14, 2002.
- [2] D. P. Anderson. (2003, Aug.) nBot balancing robot. Online. [Online]. Available: <http://www.geology.smu.edu/dpa-www/robo/nbot>
- [3] Boubaker, O., "The inverted pendulum: A fundamental benchmark in control theory and robotics," in *Education and e-Learning Innovations (ICEELI)*, 2012 International Conference on , vol., no., pp.1-6, 1-3 July 2012
- [4] N. M. A. Ghani, F. Naim, and P. Y. Tan, "Two wheels balancing robot with line following capability," *World Academy of Science, Engineering and Technology*, vol. 55, pp. 634–8, 2011.
- [5] X. Ruan, J. Liu, H. Di, and X. Li, "Design and LQ control of a two-wheeled self-balancing robot," in *Control Conference, 2008. CCC 2008. 27th Chinese*, July 2008, pp. 275 –279.
- [6] K.-H. Su and Y.-Y. Chen, "Balance control for two-wheeled robot via neural-fuzzy technique," in *SICE Annual Conference 2010, Proceedings of*, Aug. 2010, pp. 2838 –2842.
- [7] T. Nomura, Y. Kitsuka, H. Suemistu, and T. Matsuo, "Adaptive backstepping control for a two-wheeled autonomous robot," in *ICROSSICE International Joint Conference*, Aug. 2009, pp. 4687–92.
- [8] G. M. T. Nguyen, H. N. Duong, and H. P. Ngyuen, "A PID backstepping controller for two-wheeled self-balancing robot," in *Proceedings for the 2010 IFOST*, 2010.
- [9] J. Kuruvilla, J. Abraham, Midhun S, R. Kunnath, R. R. Paul, "Two wheeled self balancing robot for autonomous navigation," *International Journal of Electrical Engineering and Technology (IJEET)*, Volume 5, Issue 8, August (2014), pp. 13-20
- [10] Junfeng Wu, Wanying Zhang, and Shengda Wang, "A Two-Wheeled Self-Balancing Robot with the Fuzzy PD Control Method," *Mathematical Problems in Engineering*, vol. 2012, Article ID 469491, 13 pages, 2012.
- [11] Hau-Shiue Juang; Kai-Yew Lum, "Design and control of a two-wheel self-balancing robot using the arduino microcontroller board," in *Control and Automation (ICCA)*, 2013 10th IEEE International Conference on , vol., no., pp.634-639, 12-14 June 2013
- [12] Bell, J. & Stol, K., "Tuning a GPS/IMU Kalman Filter for a Robot Driver," *Proc. Australasian Conf. on Robotics and Automation*, Auckland, New Zealand, 6-8 Dec, 2006
- [13] Y. Oshman and I. Shaviv, "Optimal tuning of a Kalman filter using genetic algorithms," in *Proc. of AIAA Guidance, Navigation, and Control Conference and Exhibit*, 2000
- [14] T. O. Ting, Ka Lok Man, Eng Gee Lim, and Mark Leach, "Tuning of Kalman Filter Parameters via Genetic Algorithm for State-of-Charge Estimation in Battery Management System," *The Scientific World Journal*, vol. 2014
- [15] Zul Azfar, A.; Hazry, D., "A simple approach on implementing IMU sensor fusion in PID controller for stabilizing quadrotor flight control," in *Signal Processing and its Applications (CSPA)*, 2011 IEEE 7th International Colloquium on , vol., no., pp.28-32, 4-6 March 2011
- [16] Hau-Shiue Juang; Kai-Yew Lum, "Design and control of a two-wheel self-balancing robot using the arduino microcontroller board," in *Control and Automation (ICCA)*, 2013 10th IEEE International Conference on , vol., no., pp.634-639, 12-14 June 2013
- [17] Grimble, M.J., *Robust Industrial Control: Optimal Design Approach for Polynomial Systems*, Prentice Hall, 1994
- [18] *Kalman Filtering*, MATLAB version R2015a. Natick, Massachusetts: The MathWorks Inc., 2015
- [19] D. E. Goldberg, *Genetic Algorithms in Search, Optimization, and Machine Learning*, vol. 412, Addison-Wesley Reading, Menlo Park, Calif, USA, 1989
- [20] *MATLAB version R2015a*. Natick, Massachusetts: The MathWorks Inc., 2015