CEAI, Vol.21, No.4 pp. 22-33, 2019 Printed in Romania

Genetically Optimized ANFIS-based PID Controller Design for Posture-Stabilization of Self-Balancing-Robots under Depleting Battery Conditions

O. Saleem[1], M. Rizwan[2], P. S. Shiokolas[3], B. Ali[4]

[1]Department of Electrical Engineering, National University of Computer and Emerging Sciences, Lahore, Pakistan (e-mail: omer.saleem@nu.edu.pk)
[2]Department of Mechatronics and Control Engineering, University of Engineering and Technology, Lahore, Pakistan (e-mail: mohsin.rizwan@uet.edu.pk)
[3]Department of Mechanical and Aerospace Engineering, University of Texas at Arlington, Arlington, TX, 76010, USA (e-mail: shiokolas@uta.edu)
[4]School of Engineering and Applied Sciences, University of Sussex, East Sussex, England (email: bali@sussex.ac.uk)

Abstract: It is well known that in battery-powered control applications, the continuous drop in battery's output-power progressively degrades the dynamic performance of the system. The battery depletion phenomenon deteriorates the reliability of correctional effort if the controller gains are not adaptively adjusted as function of the available power level. Hence, this paper presents an adaptive neuro-fuzzy inference system (ANFIS) that dynamically adjusts the controller gains of a close-loop dynamic system as function of battery power-level in order to maintain desired performance while the battery is depleting. The proposed methodology is verified on an inherently unstable two-wheeled self-balancing-robot. The Proporional-Integral-Derivative (PID) controller is used for robot's posture-stabilization. İnitially, trivial sets of PID gains are selected via genetic algorithm to yield best control effort at various battery power-levels, using hardware-in-the-loop strategy. The acquired data is then used to train a power-level dependedent ANFIS that dynamically adjusts the PID gains in real-time. The performance of a fixed gain PID controller is compared with that of the proposed self-tuning PID controller for two different power-depletion scenarios that emulate real-world situations. The corresponding experimental results validate the robustness of the proposed control scheme to maintain the robot's postural stability under discharging battery condition.

Keywords: Battery power depletion, ANFIS, self-balancing-robot, PID controller, genetic optimization.

1. INTRODUCTION

The Direct-Current (DC) batteries are used to supply electric power at places where conventional alternating-current (AC) sources are not available or viable. They provide a reliable source of electrical energy when they are fully, or reasonably, charged. However, as the battery discharges, the available power supply capacity reduces which adversely affects the performance of the device being powered. The power depletion phenomenon is extremely detrimental to the battery-powered devices such as hybrid electric vehicles, drones, wheeled mobile robots, medical assistive devices, etc. Therefore, the knowledge of battery power depletion-rate, and its effect on the performance of the system being powered must be considered as an integral component of remedial actions in order to ensure that the system maintains the desired performance. A preliminary study performed by the authors clearly demonstrated that the available power level for control purposes affects the performance of the system. The experiments were performed on a Two-Wheeled-Self-Balancing-Robot (TWSBR) with fixed Proportional-Integral-Derivative (PID) controller gains defined for full power availability of the battery. The responses corresponding to the variations in body-angle of TWSBR for three different battery power-levels (15 W, 30 W, 50 W) are presented in Fig. 1. The responses clearly show that for certain power levels (in this case, 30 W) the system could be controlled but not within desired performance specifications. Furthermore, there exists a threshold power level (in this case, 15 W) below which the system cannot maintain the desired performance and becomes unstable.

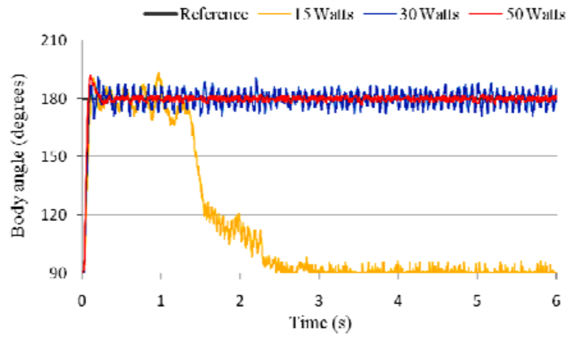


Fig. 1. Response of TWSBR with drop in motor input power with fixed PID gains.

1.1. Related work

Extensive research has been performed related to battery power management and relevant control strategies in dynamic systems. The research literature addresses topics such as development of mathematical models for battery behavior, energy management in hybrid vehicles, energy management in wind turbines, optimization of battery - super capacitor storage system, optimal energy management for an aging battery, and risk management strategies for a depleting battery. An empirical model of battery behavior during discharge cycles was developed in (Saha and Goebel, 2009; Dalal et al., 2011). A review article discussed optimal energy management strategies considering fuel consumption, emissions and economy (Panday and Bansal, 2014). A particle swarm based optimal energy management for hybrid wind-micro-turbines with multiple constraints is discussed in (Pourmousaviv et al., 2010). Control strategies for plug-in hybrid vehicles are discussed in (Wirasingha and Emadi, 2011). Reduction of investment and operating cost in hybrid energy storage station is discussed in (Zhou and Sun, 2014). Improved annealing particle swarm optimization is used to optimize the problem with constraints. Optimal energy storage control strategy for grid connected micro-grids is discussed in (Malysz et al., 2014). Power management for an aging battery used Pontryagin's minimum principle to find a solution with a compromise between aging and performance (Serrao et al., 2011). Energy management for plug-in hybrid electric vehicle using a combined "rule based and particle swarm optimization algorithm" showed improvements in energy savings compared to traditional blended strategy (Chen et al., 2015). Risk-gain based energy management for self-docking mobile robots where the battery depletion is considered as a risk and accomplishment of task as a gain and an arbitrator makes the final decision based on the assessment of risk and the level of associated accomplishment (Berenz and Suzuki, 2011).

The performance of the conventional model-based controllers is prone to be deteriorated by modelling and identification errors associated with the system. Hence, in this research, a ubiquitous PID controller is used as TWSBR's primary stabilization controller. Tuning the PID gains, K_p, K_i and K_d, is essential to maintain the desired system performance. Recently, the Genetic Algorithms (GAs) have gained a lot of momentum in optimally selecting the PID gains (Gurban et al., 2014; Vijayakumar and Manigandan, 2016). The GA is preferred over classical tuning methods, such as Ziegler-Nichols or Cohen-Coon, because it stochastically evolves and minimizes the objective function to quickly converge to optimum values of K_p, K_i and K_d (Jaen-Cuellar et al., 2013). The GA neither relies on the system model nor on the derivative of the objective function. The genetic optimization of PID controllers for various industrial processes and its comparison with classical tuning techniques has been rigorously discussed in the literature (Yusoff et al., 2015; Hussain et al., 2014). Extensive research has been done to validate the usage of genetic algorithms for the optimization of robot motions controllers and mechatronic systems (Halal and Dumitrache, 2006; Stan et al., 2007; Elbori el al., 2018). Owing to its attributes, the GA is used in this research to optimally estimate the PID controller gains for discrete levels of available battery power. The acquired data is then used to train an online power-dependent gain adjustment system that modifies the controller gains to maintain the desired system performance, even if battery is continuously discharging. The intelligent online gain adaptation systems are widely used to compensate intrinsic and un-modeled nonlinearities associated with complex dynamical systems (Chopra et al., 2014). The conventional Fuzzy-Inference-System (FIS) utilizes qualitative logical rules to infer correct control decisions. However, the finite number of rules is not sufficient to address the parametric uncertainties. The Artificial-Neural-Networks (ANNs) can derive accurate nonlinear input-output relationship due to their inherent learning capability (Mossad and Salem, 2014). However, their synthesis requires large sets of training data. The Adaptive-Neuro-Fuzzy-Inference-System (ANFIS) synergistically combines the learning capability of ANNs with reasoning-based inference of FIS to deliver robust adaptive control effort (Ioanaş, 2012; Cărbureanu, 2014). Wherein, the ANNs serve to automatically update the parameters of the rule-based FIS to enhance the system's robustness against exogenous disturbances (Szymak, 2016). The ANFIS quickly develops an accurate numerical model to effectively control nonlinear dynamical systems with minimal training data (Kumar et al., 2015).

The available literature mainly discusses energy management strategies for either overall improved performance or a favourable compromise between risk and gain while the battery is depleting. In open literature, the dynamic adjustment of PID gains as function of available power level, as the battery is discharging, is not discussed. The aforementioned idea is the main focus of this research article.

1.2. Proposed approach

This research presents a platform to examine and adaptively compensate the effects of power depletion and remaining power on the performance of a dynamical system. Hence, the main contribution of this article is to synthesize a suitable online gain adjustment law that effectively rejects the detrimental effects rendered by battery depletion on posture stability of TWSBR, apart from other exogenous disturbances. The proposed technique utilizes a well-postulated GA to generate reference-data in order to train a dedicated ANFIS model. The derived data-model is then used to dynamically update the PID gains of TWSBR's stabilization controller, with respect to the variations in battery power conditions, after every sampling interval. This augmentation improves the controller's robustness by maintaining the robot's postural stability while rejecting the influence of continuously reducing battery power. The proposed adaptation scheme recursively computes the error and minimizes the difference between the actual and desired system outputs. The ANFIS is trained off-line using the optimized reference data set generated by the GA. Once the ANFIS is completely trained and incorporated with the stabilization controller, it serves to dynamically adjust the gains (K_p, K_i, K_d) and applies them to the PID controller. To conduct the hardware-in-the-loop experiments, the power variations observed during battery depletion are emulated via a DC-DC buck converter. The proposed adaptive controller can only effectively stabilize the posture of the TWSBR for a range of available battery power. The lower bound of this range, for this research, is 15.0 W.

The organization of the remaining paper is as follows. The dynamical model of the overall system is presented in Section 2. The primary feedback control scheme is discussed in Section 3. The training-data acquisition methodology is discussed in Section 4. Detailed synthesis of ANFIS model is presented in Section 5. The experimental setup is explained in Section 6. The experimental evaluation is conducted in Section 7. The paper is concluded in Section 8.

2. SYSTEM MODELING

The TWSBR is modeled as a rigid pole that is fastened to a rigid cart via a frictionless joint. The cart moves along the longitudinal axis via two coaxial motorized wheels.

2.1. Actuator dynamics

The state-space model of a linear dynamical system is generally given by (1).

$$\dot{x} = Ax + Bu, y = Cx + Du \tag{1}$$

where, x is the state-vector, y is the output-vector, u is the control input signal, A is the system matrix, B is the input matrix, C is the output matrix, and D is the feed-forward matrix. The state-vector of a DC motor is given by (2).

$$x = [\varphi \quad \omega]^T \tag{2}$$

where, φ and ω is the angular position and velocity of the wheel, respectively. Correspondingly, the matrices A, B, C, and D in the motor's state-space model are identified in (3).

$$A = \begin{bmatrix} 0 & 1 \\ 0 & \dfrac{K_b{}^2}{J_m R} \end{bmatrix}, B = \begin{bmatrix} 0 \\ \dfrac{K_b}{J_m R} \end{bmatrix}, C = [1 \quad 0], D = 0 \tag{3}$$

The motor voltage, V_m, is taken as the control input signal. The motor parameters are identified in Table 1.

2.2. Body dynamics

All the forces acting on the robot body must be considered while deriving TWSBR's dynamic model. The free-body diagram of the right-wheel is shown in Fig. 2(a). Both wheels have similar dynamic properties. The wheel structure is assumed to be uniform and homogenous, allowing it to roll over the horizontal surface without slippage. The combined effect of both motors in moving the body is given in (4).

$$2\left(m_w + \frac{J_w}{r^2}\right)\ddot{x} = 2\left(\frac{K_b}{Rr}\right)V_m - 2\left(\frac{K_b{}^2}{Rr^2}\right)\dot{x} - (H_R + H_L) \tag{4}$$

where, m_w is the mass of the right-wheel, $\ddot{x}$ is the longitudinal acceleration of the right-wheel, $H_{R,L}$ is the z-axis force of the right- or left-wheel with the robot's body, J_w is the Moment-of-Inertia (MoI) of the wheel, τ_R is the wheel torque, H_{fR} is the frictional force, and r is the radius of the right-wheel.

The robot's upper body is modelled as the arm of an inverted pendulum. Its free-body diagram is shown in Fig. 2(b). The sum of horizontal forces acting on the body is given by (5).

$$\sum F_x = m_p\ddot{x} = H_R + H_L - m_p l\ddot{\theta}\cos\theta + m_p l\dot{\theta}^2\sin\theta \tag{5}$$

where, m_p is the mass of the robot chassis, θ is the angle subtended between the robot and the vertical-axis (also known as the pitch-angle), and l is the length of the Centre-of-Mass (CoM) of body from the ground. The continuous-time state-space representation of the TWSBR system is given by (6) and (7), respectively.

$$\begin{bmatrix} \dot{x} \\ \ddot{x} \\ \dot{\theta} \\ \ddot{\theta} \end{bmatrix} = \begin{bmatrix} 0 & 1 & 0 & 0 \\ 0 & a_1 & a_2 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & a_3 & a_4 & 0 \end{bmatrix} \begin{bmatrix} x \\ \dot{x} \\ \theta \\ \dot{\theta} \end{bmatrix} + \begin{bmatrix} 0 \\ b_1 \\ 0 \\ b_2 \end{bmatrix} V_m \tag{6}$$

$$y = [0 \quad 0 \quad 1 \quad 0] \begin{bmatrix} x \\ \dot{x} \\ \theta \\ \dot{\theta} \end{bmatrix} \tag{7}$$

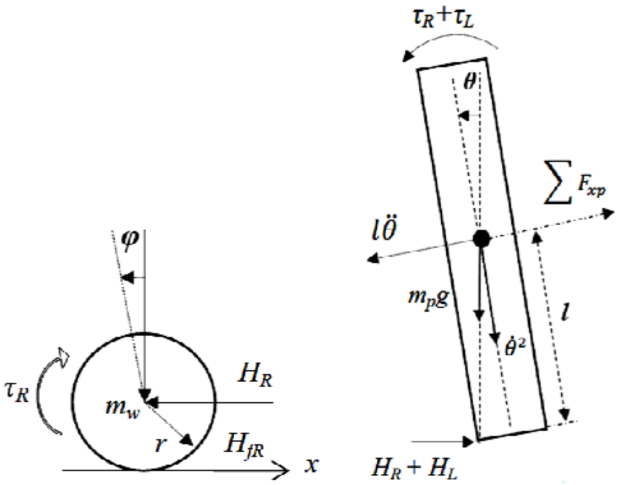
where,

$$a_1 = \frac{2K_b{}^2(m_p lr - J_p - m_p l^2)}{Rr^2\gamma}, a_2 = \frac{m_p{}^2 g l^2}{\gamma}, a_3 = \frac{2K_b{}^2(J_p + m_p l^2 - m_p l)}{Rr^2\gamma},$$

$$a_4 = \frac{m_p g l\beta}{\gamma}, b_1 = \frac{2K_b(r\beta - m_p lr)}{Rr\gamma}, b_2 = \frac{2K_b(m_p l - r\beta)}{Rr\gamma},$$

$$\beta = 2m_w + m_p + \frac{2J_w}{r^2}, \gamma = J_p\beta + 2m_p l^2\left(m_w + \frac{J_w}{r^2}\right)$$

The system parameters are identified in Table 1.



(a) (b)

Fig. 2. Free body diagram of (a) the right wheel, (b) the robot body.

Table 1. System parameters of TWSBR.

Parameter	Symbol	Value
Motor MoI	J_m	2.18×10^{-4} kgm^2
Motor torque constant	K_b	0.073 Nm/A
Motor Armature resistance	R	3.3 Ω
Gravitational Acceleration	g	9.81 m/s^2
Wheel radius	r	0.121 m
Wheel mass (with motor)	m_w	0.258 kg
Wheel MoI	J_w	1.87×10^{-3} kgm^2
Body CoM height	l	0.158 m
Body mass	m_p	1.85 kg
Body MoI	J_p	0.0247 kgm^2

2.3. LiPo battery discharge dynamics

In this research, the TWSBR is powered by a Lithium Polymer (LiPo) battery. The circuit diagram of the LiPo battery cell is shown in Fig. 3. The piecewise nonlinear equations presented in (8) demonstrates the discharging behavior of a LiPo battery (Kim et al., 2013).

$$V_{terminal}(t) = \begin{cases} V_{oc}(t) - I(t)R_{dc} - I(t)R_s\left(1 - e^{-\frac{(T-t)}{R_sC_s}}\right), t \le T \\ V_{oc}(t) - I(t)(R_{dc} + R_s) - I(t)R_l\left(1 - e^{-\frac{(T-t)}{R_lC_l}}\right), t \le T - \tau_s \\ V_{oc}(t) - I(t)(R_{dc} + R_s + R_l), t \le T - \tau_s - \tau_l \end{cases} \quad (8)$$

such that, $T = m \times C_b, t = SOC(t) \times C_b, \tau_s = R_sC_s, \tau_l = R_lC_ls$

where, I is the current provided by the cell, SOC is the state-of-charge, and m is the initial SOC of the battery before discharge, C_b is the capacity of battery cell, R_d is the resistance of self-discharge, V_{oc} is the open-circuit voltage of the battery cell, R_{dc} is the equivalent DC-resistance of the cell, R_s and C_s is the resistance and capacitance of the short time-constant RC circuit in the model, respectively, R_l and C_l is the resistance and capacitance of the short time-constant RC circuit in the model, respectively. When the discharging initiates at $t = T$, the voltage across the battery terminals ($V_{terminal}$) is lesser than the V_{oc} due to the voltage-drop across R_{dc} and the exponential voltage-drop across the short time-constant RC circuit, for $t \le T$. The voltage continues to drop for the duration of τ_s. Afterwards, the long time-constant RC circuit contributes in the exponential voltage-drop for $t \le T - \tau_s$. It keeps dropping for the duration of τ_l. Finally, the voltage drop becomes purely resistive and is significantly larger than the previous cases (Kim et al., 2013).

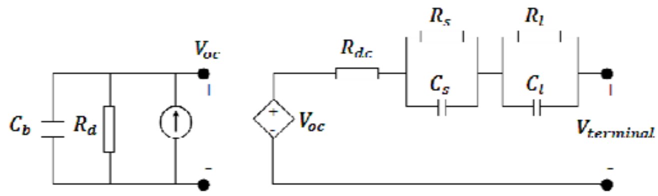


Fig. 3. LiPo battery cell circuit model.

3. FEEDBACK CONTROL SYSTEM

This section discusses the feedback control architecture of the system. A PID control scheme is employed for posture stabilization of TWSBR (Li et al., 2013). The posture of TWSBR is stabilized by continuously monitoring and comparing the variations in the robot's body-angle with the reference-angle using an Inertial-Measurement-Unit (IMU). The resulting errors are fed to a digitally implemented PID controller that appropriately drives the motors and keeps the robot vertically balanced. Owing to its simplicity and reliability, the PID scheme is widely used as the primary robot motion controller (Astrom and Hagglund, 2016). In this research, it is used to regulate the pitch-angle, ϕ, of TWSBR's body within desired performance characteristics (Bhatti et al., 2015). The horizontal motion (or position) of TWSBR is not controlled in this research. PID controller is simply the weighted sum of error-dynamics occurring in the state-variable being controlled; namely, error, error-integral, and error-derivative (Bhatti et al., 2018). The error, $e_\theta(n)$, at sample time n is defined as the difference between the desired set-point, $\theta(des, n)$, and actual, $\phi(n)$, states as shown in (9). The mathematical expression of the PID control law for hardware implementation is given by (10).

$$e_\theta(n) = \theta(des, n) - \phi(n) \quad (9)$$

$$u(n) = u(n-1) + K_pe_\theta(n) + K_d\frac{e_\theta(n) - e_\theta(n-1)}{T_s} + K_i\sum_{j=1}^{n} e_\theta(j)T_s \quad (10)$$

where $u(n)$ is the control signal at sample time n, and T_s is the sampling period. The diagram of the closed loop control scheme is shown in Fig. 4. Initially, the gains of the PID controller are tuned using GA assuming a fully charged battery and available maximum motor power level. As the system operates, the battery power level is reduced and the PID gains are optimally tuned via GA for a set of discrete levels of actuator or motor power to generate a reference data-set of gain values as function of available power level. In this research, 100 discrete power levels were selected. However, one could select a different number depending on the application and desired granularity for available power level. After setting the motor power at a certain level, the GA is initiated to identify the PID gains such that the system operates satisfactorily and remains within acceptable performance specifications, i.e. close to the desired vertical posture.

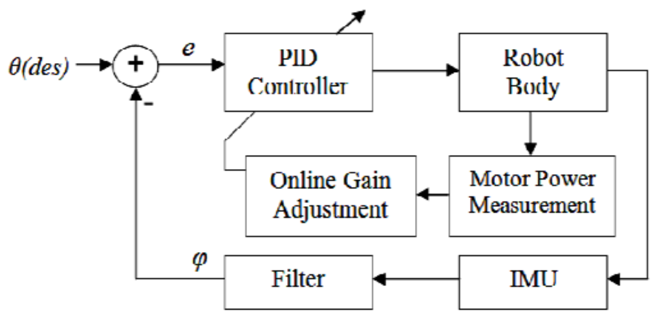


Fig. 4. Closed-loop feedback control scheme.

Once the algorithm converges, the optimized PID gains for the given power level are recorded. The process of tuning the PID gains is repeated for each discrete power level. The PID gains recorded for each power level are then used to synthesize mathematical models for each gain as function of available power. The recorded data is used to train the

ANFIS model that dynamically updates the PID gains to deliver appropriate control actions, *u(n)*, as function of available power level to maintain desired system performance.

4. TRAINING-DATA ACQUISITION USING GA

The optimal sets of PID gains must be identified at discrete levels of battery power to derive an accurate numerical model for training the ANFIS-based online gain adjustment law as function of available power level using the GA. The GA is a population-based stochastic search and optimization algorithm based on natural evolution (Petcut and Dragomir, 2010; Mohammed at al., 2014). It probabilistically explores a randomly selected population (or search-space) of candidate solutions to find the global-best solution (Tam et al., 2018). After the initialization of the search space, the algorithm encodes all potential solutions into strings of binary numbers, also referred to as "chromosomes", and evaluates their performance using an objective function that assigns a performance based fitness value to all the strings in a population. The highly fit strings are selected, paired and mutated to form a new population of candidate solutions. The parent strings mate with each other, via the process of "cross-over", to generate off-springs with relatively higher fitness. A few strings in the population are mutated to ensure a uniformity of fitness in the entire population and avoid the possibility of new stagnant or unsuitable populations (stuck in local minima regions). After every iteration of reproduction, the succeeding generations tend to evolve the search space with strings having improved fitness values. Thus, populating the space with high-ranking solutions enables the algorithm to converge quickly to the global-best solution. The process is repeated iteratively and is terminated only if either the desired performance criterion is achieved or the pre-defined maximum number of iterations is reached. The workflow of GA is shown in Fig. 5.

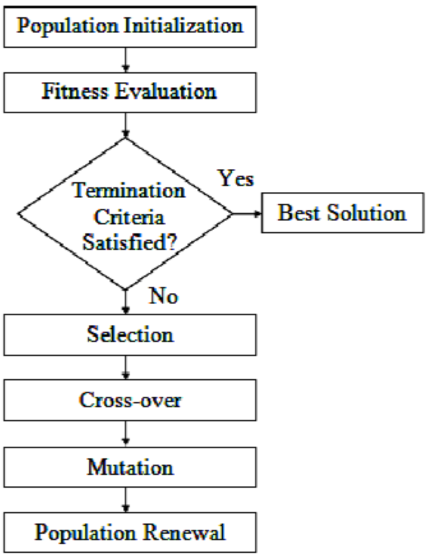


Fig. 5. Work-flow of GA procedure.

In this research, the GA generates an initial random population by encoding the PID gains into binary strings (or chromosomes). For a given power level, each trivial set of gain is fed to the PID controller and the corresponding body-angle response of the robot is recorded for 10 seconds (1000 samples). The resulting fitness of each population member is evaluated after decoding the binary string into real-valued PID gains and applying them to the actual hardware. In this work, considering the objective of keeping the robot vertically balanced, the cost function is defined as the sum of squared-error within the defined time-span for every individual experiment. This process is repeated until the defined number of generations is reached in an effort to reach near-optimal values of the PID gains meeting the desired performance specification of balancing the robot at a given power-level while minimizing the fitness function.

The details and steps followed to tune the PID controller gains for each motor-power level are discussed as follows.

Initialization: An initial population of 50 strings is selected to represent the three PID gains. The gains in the population are bounded within the interval 0 to 30, considering the convergence point in repeated trial. Each string is composed of 30 bits. The length of string is divided in three sections; each containing a sub-string of 10 bits with each distinctly corresponding to one of the three PID gains.

Fitness evaluation: In this research, the objective function, J, to be minimized is defined as the sum-of-squared-error. Its mathematical expression is given by (11).

$$J = \sum_{n=1}^{N} (e_\theta(n))^2 \tag{11}$$

where, n is the sample number and N is defined as 1000 for all cases irrespective of how quickly the parameters converge. This objective function is selected as the criterion to evaluate the fitness of each PID gain string in the population. Every time a string is subjected to the objective function, the string is divided to extract three PID gains which are directly applied to the robot's PID control law. The corresponding control commands are serially transmitted to the motor driver circuit of the robot. The variations in the error of the robot's body-angle are recorded to evaluate the fitness of the particular string.

Selection: The number of strings selected to form the subspace must be decided carefully; a large subspace may deliver a "better" solution but it would lead to a slower convergence rate, where, a smaller subspace would speed up the evolution process, but it may lead to premature convergence and possibly deliver bad solutions. In this work, 12 best (highly fit) strings are selected from the existing population and another 20 strings are randomly selected from the remaining 38 strings, thus generating a subspace of 32 strings as parents.

Crossover: The selected parent strings are then combined via the multi-point crossover operator, wherein the alternating segments of the corresponding subsections of parent strings are swapped to generate new off-springs. The crossover operator is applied with a probability of 0.7. Each new offspring is compared with the existing worst-string and if the off-spring has better fitness, then it is retained; otherwise, it is discarded.

Mutation: The evolved population is slightly mutated in order to prevent the algorithm from falling and possibly remaining in local minima regions and to improve the performance of the strings. The mutation operator is applied with a low probability of 0.063. One or more randomly selected bits of the chosen strings are flipped.

Termination: Once the new population is formed, it is analyzed for the termination criterion. The algorithm is only terminated when the difference between the fitness values of highest-ranked and the lowest-ranked strings in a population is less than or equal to the fitness limit (β), as shown in (12).

$$\left|J_{i,best} - J_{i,worst}\right| \leq \beta \tag{12}$$

where, i is the current iteration (or generation) and J is the fitness value of a given string in a given population. In this research, the value of β is empirically selected as 0.1 to ensure convergence and to guarantee that all the candidate solutions are closely spaced. Once the proposed criterion is satisfied, the algorithm is terminated and the best string is selected as the output. Otherwise, the process is repeated until a maximum number of iterations are reached to avoid the possibility of non-convergence or an infinity loop of execution. The process of iterative genetic optimization of the PID gains (K_p, K_i, K_d) at a power level of 50.0 W is shown in Fig. 6. The training data of the K_p, K_i, and K_d generated by the GA at 100 discrete power levels is diagrammatically illustrated in Fig. 7, 8, and 9, respectively.

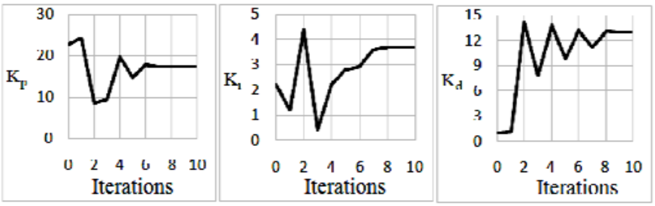


Fig. 6. Genetic optimization of PID gains at 50.0 W level.

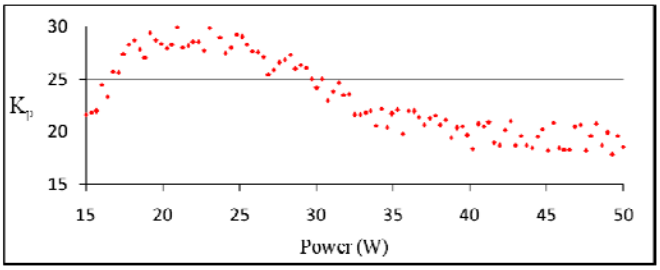


Fig. 7. Genetically optimized training data of K_p.

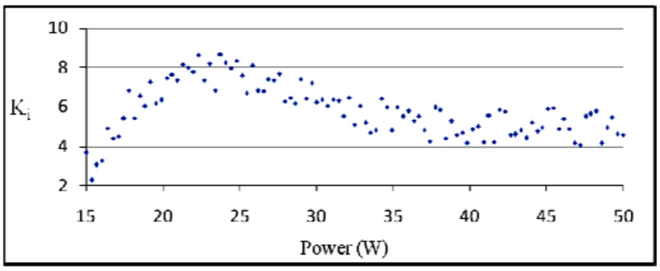


Fig. 8. Genetically optimized training data of K_i.

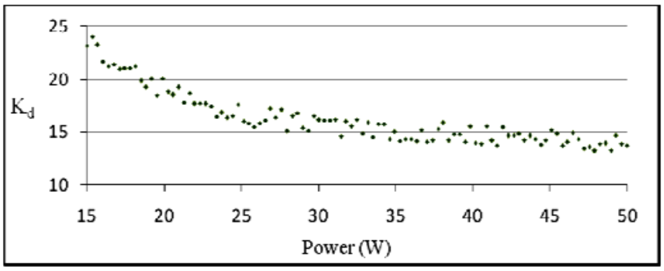


Fig. 9. Genetically optimized training data of K_d.

5. PROPOSED ONLINE GAIN-ADJUSTMENT SCHEME

Initially, the PID controller gains are tuned assuming a fully charged battery. However, as the battery discharges, the DC input power to the motors drops. Without adaptive control, this reduction in the available power reduces the motor's ability to deliver the required dynamic actuation effort. The body angle of the robot, as function of time, for different power levels is already shown in Fig. 1. At reduced power, the robot begins to vibrate vigorously about its reference position. At ~15.0 W, these vibrations increase and the system eventually collapses, as shown in Fig. 1.

The performance degradation rendered by reduced available power can be compensated by dynamically scheduling the PID gains as a function of available battery power using the ANFIS, which is an adaptive realization of FIS that utilizes a multi-layered feed-forward network (Toufouti et al., 2009). First of all, a two-dimensional table of heuristically developed fuzzy rules is identified. The hypothesis of the parameterized rule-base serves to relate the fuzzified inputs to input Membership-Functions (MFs) to rules to outputs to output MFs. In the next phase, the optimized data set acquired by GA is applied to train the ANFIS. The trained ANFIS is then used to formulate the fuzzy inference mechanism by optimally adjusting its MFs such that they fulfill the desired control objective(s). The ANFIS imitates a feed-forward back-propagation network. In this research, the neural part of the ANFIS is trained using the Hybrid-Learning-Algorithm (HLA). In the first phase of training, the HLA identifies the consequent parameters in the forward pass with the aid of Least-Squares method. In the second phase of training, the HLA it uses the error between the actual and desired system output as well as the gradient descent method to update the premise parameters in the backward pass. The training process terminates only when the maximum number of iterations is reached or the desired control objective is achieved**.** The fuzzy system is developed using the first-order Takagi-Sugeno inference model due to its high interpretability and computational efficiency.

The conventional five-layered architecture of ANFIS is shown in Fig. 10. The layers are denoted as the fuzzy layer, product layer, normalization layer, defuzzification layer, and total output layer. In this structure, the error in battery power (P_e) and the change in power-error (ΔP_e) are the inputs. The inputs are given by (13) and (14).

$$P_e(n) = P_{max} - P(n) \tag{13}$$

$$\Delta P_e(n) = P_e(n) - P_e(n-1) \tag{14}$$

where, P_{max} is equal to 50.0 W in this research, The vector containing PID gain updates, denoted as K_z, is the output of the proposed ANFIS model. The training values and the estimated values are represented by the input and output nodes, respectively. The nodes in the hidden layers operate as the MFs and rules. The nodes represented by a circle are fixed-nodes while the ones represented by a square shape are adaptive in nature. For a first-order Sugeno inference system, the governing rules are as follows:

Rule 1: If $g=P_e$ *is* A_1 *and* $h=\Delta P_e$ *is* B_1, *then* $K_{z1} = p_1 g + q_1 h + r_1$

Rule 2: If $g=P_e$ *is* A_2 *and* $h=\Delta P_e$ *is* B_2, *then* $K_{z2} = p_2 g + q_2 h + r_2$

where, A_j and B_j are the linguistic variables of the fuzzy rule-base, p_j, q_j, r_j are the consequent parameters, and j is the number of rule being considered (j = 1 or 2). The function of each layer is as follows (Saleem et al., 2018). The output of node i in layer l is denoted as O_i^l.

Layer 1 is the fuzzification layer. It fuzzifies the inputs g and h using the bell-shaped MFs due to their smoothness. The mathematical expression of MF is given by (15).

$$O_i^1 = \frac{1}{1 + \left|\frac{x-c_i}{a_i}\right|^{2b_i}} \tag{15}$$

where, a_i, and c_i are the width and the center of the MF, respectively, and b_i is used to control the slopes of the cross-over points. These premise parameters are dynamically adjusted via the ANN to yield optimum postural stability with minimum tracking deviations, even under depleting battery conditions. Every node in this layer is adaptive.

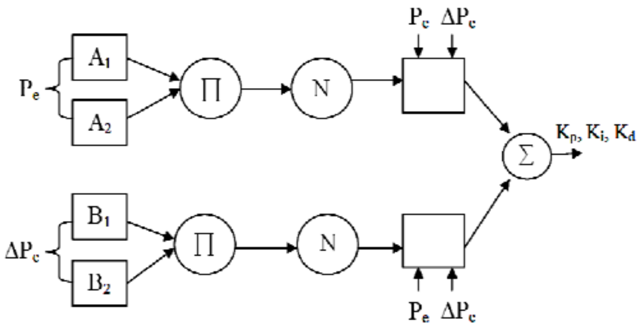


Fig. 10. The ANFIS structure.

The layer evaluates the degree (μ) of input variables in the fuzzy set. The output of this layer is $\mu_{Aj}(g)$ or $\mu_{Bj}(h)$ and are given by (16) and (17).

$$O_i^1 = \mu_{Ai}(g), \qquad such\ that\ i = 1, 2 \tag{16}$$

$$O_i^1 = \mu_{Ai}(h), \qquad such\ that\ i = 3, 4 \tag{17}$$

The output of this layer forms the antecedents of 2nd layer. Once the consequent parameters are selected (see Layer 4), the approximate error is propagated back to every layer and the gradient-descent method is used to update the premise parameters.

Layer 2 is denoted as the product layer. It mathematically infers the updates required in the PID gains. Every node in this layer is fixed and corresponds to a fuzzy antecedent rule (the "*IF*" part). A total of 25 rules are used in this layer. The product T-norm aggregation operator is used with each fuzzy neuron. Hence, the output of this layer is the product of the input signals as shown in (18).

$$O_i^2 = w_i = \mu_{Ai}(g) \times \mu_{Bi}(h) \tag{18}$$

The resulting weightage, w, represents the firing-strength of each rule. The fuzzy MFs of the input-variables, P_e and ΔP_e, and output-variables, K_p, K_i, K_d, are linguistically defined as: Zero (Z), Small (S), Medium (M), Big (B), and Very Big (V). The two-dimensional rule-base is shown in Table 2.

Layer 3 is the normalization layer. Every node in this layer is fixed. This layer serves to normalize the firing strengths by evaluating the ratio of the i^{th} fuzzy-rule's firing strength to sum of all firing strengths of all the rules. The normalized firing strength is given by (19).

$$O_i^3 = \widehat{w}_i = \frac{\sum w_i}{\sum_i w_i} \tag{19}$$

Layer 4 is defined as the consequent layer. It is responsible for defuzzifying the consequent rules. The nodes in this layer are adaptive and they compute the values of i^{th} rule towards the overall output (the "*THEN*" part) using the linear combination shown in (20).

$$O_i^4 = s_i = \widehat{w}_i\,(p_i g + q_i h + r_i) \tag{20}$$

Table 2. Fuzzy rule base of PID controller gains.

K_p,K_i,K_d		ΔP_e				
		Z	**S**	**M**	**B**	**V**
P_e	**Z**	Z,Z,Z	Z,S,S	S,S,M	S,Z,B	S,Z,V
	S	S,M,S	S,M,M	S,S,B	S,Z,V	M,Z,V
	M	M,M,M	M,B,M	M,B,B	M,S,V	M,Z,V
	B	B,S,M	B,S,M	B,M,B	B,S,V	B,Z,V
	V	V,S,M	V,S,M	V,M,B	V,S,V	V,Z,V

The consequent parameters are adaptively adjusted to yield best control effort, even under battery discharge conditions. Since O_i^4 is the linear combination of the consequent parameters, therefore, their optimal values are estimated via Least Squares method in the backward pass of training phase.

Layer 5 is summation layer. The single node in this layer computes the crisp output value by aggregating all the incoming signals from Layer 4. The finalized output is given by (21).

$$O_i^5 = \sum_i s_i = \frac{\sum_i w_i\,(p_i g + q_i h + r_i)}{\sum_i w_i} \tag{21}$$

The ANFIS is digitally implemented using a 64-bit, 2.6 GHz, and 6GB RAM computer. Owing to the processing power of computer, the scheme does not put excessive computational burden on the digital computer. The practicability of the proposed adaptation scheme is validated in Section 7, where it is applied to adaptively tune the PID controller of an actual TWSBR system. The robotic system is subjected to hardware-in-the-loop experiments in real-time.

6. EXPERIMENTAL SETUP

The experimental setup used to verify the proposed methodology of considering power depletion and dynamic PID gain definition, while maintaining the desired performance, is described in this section.

6.1 Hardware setup

The TWSBR platform used for real-time experimentation is shown in Fig. 11. The physical dimensions of the robot body are 0.2 m × 0.2 m × 0.33 m. The robot uses an 8-bit, 5.0 V embedded microcontroller for monitoring and control tasks. The IMU contains an accelerometer and a gyroscope that provides the body-angle and its rate-of-change, respectively.

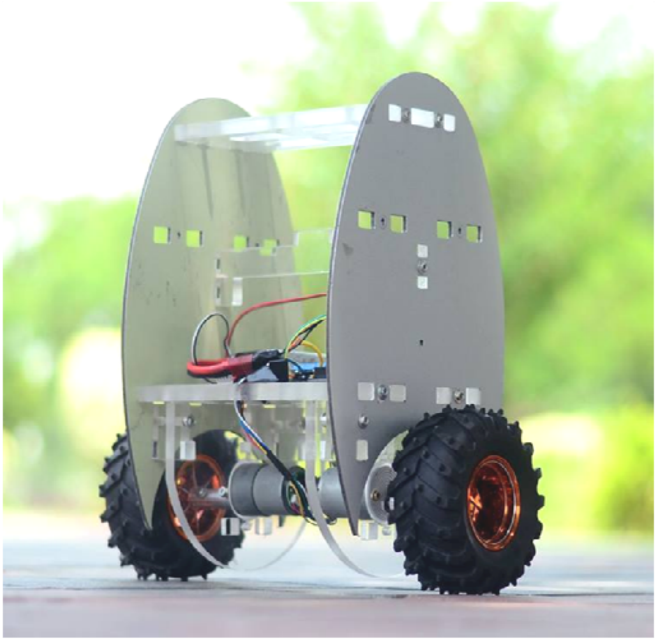


Fig. 11. Two-wheeled self-balancing robot.

Two permanent magnet DC metal-geared motors are attached beneath the base of the robot structure to drive the wheels. Each motor has a 19:1 gear box. It is capable of delivering a torque of 0.588 Nm and a non-load speed of 500 rpm. The motors are driven by MC33926 dual H-bridge motor driver.

For experimentally emulating *fast* battery power depletion, the input power to the motors is varied by controlling a 100 W programmable DC-DC step-down buck converter. The buck converter effectively alters the magnitude of the input power to the motors. Buck converters are energy-efficient switch-mode power supplies (Rashid, 2011). The block diagram of the power variation setup is shown in Fig. 12. The MOSFET in the circuit chops down the DC supply voltage to the desired level. The inductor-capacitor filter removes the harmonics from the output-voltage. An analog voltage-divider, constructed with a 2.0 MΩ and a 1.0 MΩ, 0.5W resistors, is used to measure the voltage drop across each motors. The motor current consumption is measured using ACS712 current sensor (Jamaluddin et al., 2013).

The ANFIS model uses the instantaneous input power available to the motors which is evaluated as the product of the analog current, I_s, and voltage, V_s, readings provided to the microcontroller. The variation in the input power available to the motors as well as the TWSBR's pitch-angle is visualized using a graphical user interface (GUI) developed in LabVIEW software, running on a remote computer. In this research, the buck converter is remotely controlled over a wireless link. These commands are serially transmitted at 9600 bps to the on-board microcontroller over a wireless communication link established by a 2.4 GHz transceiver. Based on the commanded input, the microcontroller compares the reference voltage with the instantaneous voltage supplied to the motors, via the H-bridge motor driver, and generates appropriate pulse-width-modulated commands to the switching transistor in the circuit. The electronic hardware setup is shown in Fig. 13.

A 12.0V, 3300 mAh LiPo battery is used to supply DC power to all the modules of the robot via dedicated linear voltage-regulator circuits. When fully charged, the battery reads 12.6V across its terminals. The maximum power drawn by the motors is 50.0 W. Preliminary experiments indicate that below 15.0 W, the motors cannot provide the appropriate control torque to keep the robot upright (see Fig. 1).

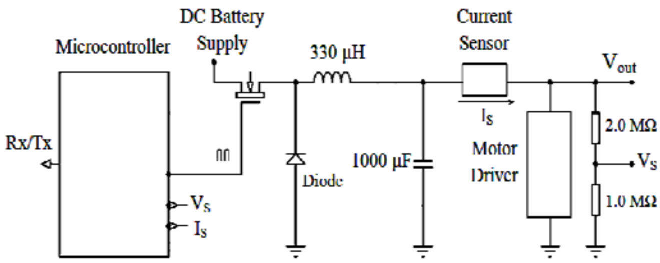


Fig. 12. Block diagram of power variation setup

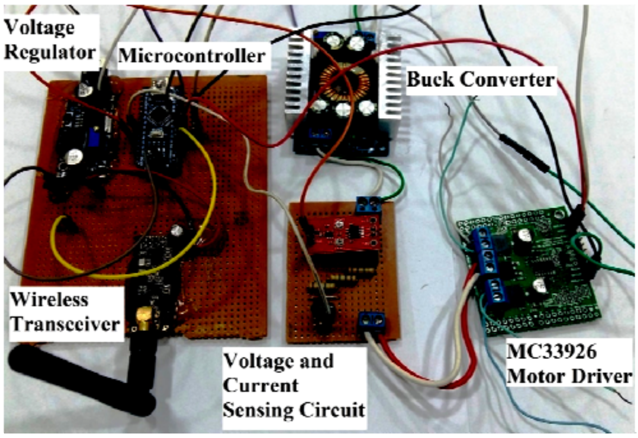


Fig. 13. Electronic hardware setup.

6.2 Control software architecture

The control software is responsible for vertically balancing the robot. Upon initialization, a Kalman Filter fuses the raw gyroscope and accelerometer readings to estimate the robot body-angle (Gui et al., 2015). The error between the actual and the desired body-angles is provided as input to the PID controller which controls the speed and direction of motor rotation. The sampling-time (T_s) for data acquisition and control updates is set at 10.0 ms. The PID control routine is programmed in the 8-bit microcontroller. The microcontroller acts as an intermittent communication relay between the TWSBR hardware and the remote computer. During the training-data acquisition phase, the GA is run on a 64-bit, 2.6 GHz computer. For a given power level, initially the GA serially transmits the optimized set of PID gains to the 8-bit embedded microcontroller and acquires corresponding measurements regarding the TWSBR's body-angle to generate the reference data. The ANFIS model is trained off-line using the recorded dataset of the tuned PID gains. Once synthesized and connected to the PID balancing controller, the ANFIS dynamically adjusts and issues the updated PID gains to the microcontroller, after every sampling interval. The corresponding voltage control commands are applied to the DC motors via the H-bridge circuit. The resulting raw analog sensor readings regarding the pitch-angle as well as the available battery power-level are acquired by the microcontroller. The microcontroller filters the sensor readings and feeds them to the ANFIS running in the computer for online gain adjustment. The proposed strategy does not put excessive computational burden on the digital computer during real-time control application.

7. EXPERIMENTAL EVALUATION

The self-balancing control of two-wheeled robot is posed as a regulator problem to keep the controlled state (pitch-angle) at the pre-defined reference value. Hence, the battery depletion phenomenon is considered as a source of exogenous disturbance to the system. The proposed control strategy is devised to reject this disturbance. Two power depletion test scenarios are considered, continuous and step depletion, to study and compare the performance of the ANFIS-based PID controller with that of a fixed-gain PID controlle to vertically balance the robot with minimum deviations. The gains of the fixed PID controller are defined at the maximum motor power level. According to Fig. 7, 8, and 9, the gains at 50.0 W are $K_p = 17.3$, $K_i = 3.7$, $K_d = 13.0$. The gains of ANFIS-based PID controller are adaptively varied as functions of the available battery power. The real-time variations in the TWSBR's pitch-angle are visualized using a LabVIEW based GUI. The robot's body-angle response are compared in terms of root-mean-square-error (RMSE) and maximum-absolute-error (MAE) in the body-angle.

7.1 Continuous power change

In this test, the motor input power is varied exponentially, as shown in Fig. 14, to emulate the actual discharging pattern of a DC battery. The error in the robot's body-angle from the vertical reference for the fixed and adaptive controllers are shown in Fig. 15 and 16, respectively. The corresponding variations in adaptively adjusted PID controller gains are depicted in Fig. 17.

As the power level decreases, the error of the fixed PID controlled system significantly increases, reaching an MAE of 14.8^o and an RMSE of 6.8^o. The adaptive controller effectively maintains the robot's postural stability, despite the continuous power decay. The adaptively controlled system exhibits an overall RMSE of only 1.5^o and an MAE of 4.7^o, which is well within the desired angular deviation of ±5.0^o in the robot body. The results clearly validate the efficacy of the proposed adaptive controller in maintaining the robot's postural stability under continuous battery discharge conditions.

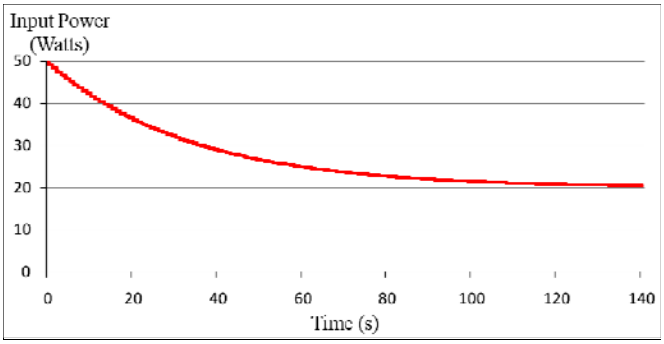


Fig. 14. Motor nput power pattern for continuous power change.

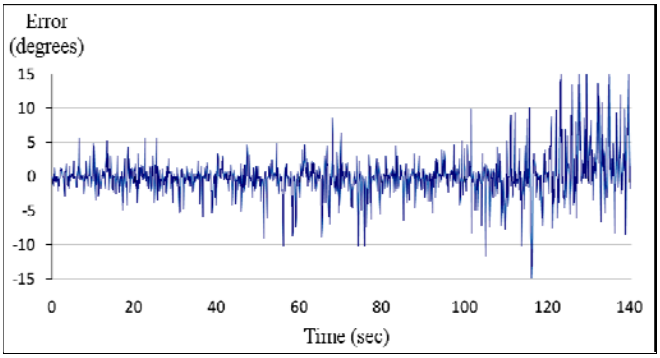


Fig. 15. Error in body-angle using generic PID controller for continuous power change.

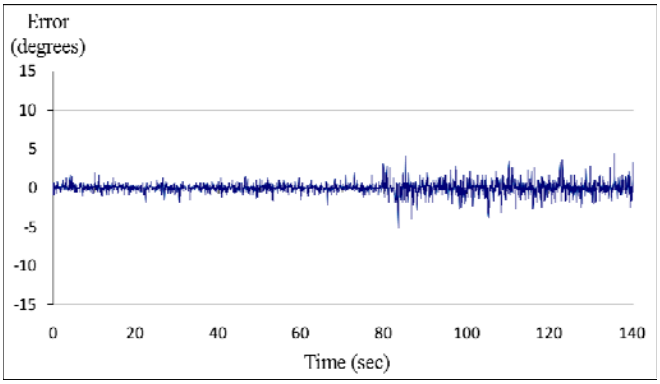


Fig. 16. Error in body-angle using the gain-scheduled PID controller for continuous power change.

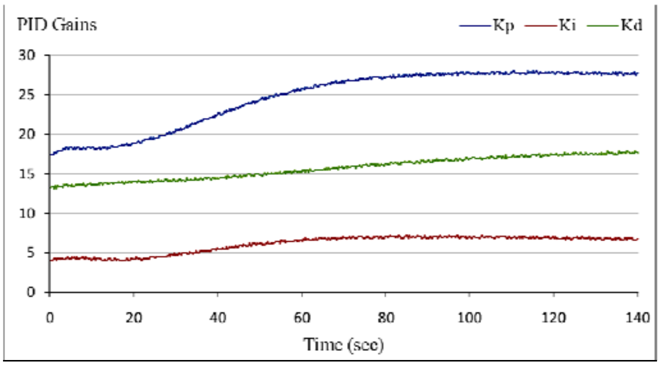


Fig. 17. Dynamic variation in PID gains using the gain-scheduled PID controller for continuous power change.

7.2 Step power change

In this test, the motor input power is varied in a step fashion as shown in Fig. 18, which is again accomplished by using the wirelessly operated programmable buck converter circuit. The power level is decreased in steps of 7.5 W at regular intervals of 10 sec. It is to be noted that this power change is arbitrarily selected. The error in the robot's body-angle from the vertical reference position, when controlled without and with the ANFIS-based gain scheduler is shown in Fig. 19 and 20, respectively. The variation pattern of dynamically adjusted PID gains, responsible for improving the TWSBR's pitch-angle response, is shown in Fig. 21. İnitially, the fixed-gain PID controller maintains the robot's vertical stability since full power is available. However, as the power level drops after fixed time intervals, the body-angle exhibits abrupt bounded variations. It is observed that the body-angle response does not converge within ±5.0^{o} of the reference position after each step-change in power level. Instead, the magnitude of the error increases reaching an MAE of 14.8^{o} and an overall RMSE of 9.1^{o}. The performance of the proposed ANFIS model to dynamically define the PID gains as function of the available power level is also evaluated. The PID gains are defined when a power change occurs and remains fixed during that particular time interval. This system exhbibits small overshoots and undershoots at the instance of step-change with an MAE of 6.8^{o} and an RMSE of 2.7^{o}. It is observed that after each power step change, the response initially has relatively a 'large error'. However, this error settles to within ±1.0^{o} of the desired performance within 2.0 to 3.0 seconds indicating that the proposed methodology can effectively compensate power transients as well. It is also observed that as the available power tends towards the limiting power level for sucessful control action (15.0 W for this system), the system exhibits a small increase in the magnitude of error compared to higher power levels, but, still remains within closer proximity of desired posture compared to the fixed gain system. This reinforces the premise of dynamically varying the gains as function of available power.

7.3 Discussion

The performances of the two PID controllers (fixed and adaptive) for the two testing scenarios of varying power level (uniform exponential decay and step) is summarized in Table 3. The time taken by the robot to settle within ±2.0^{o} of the reference position, after each step change in power-level, is denoted as T_{set}.

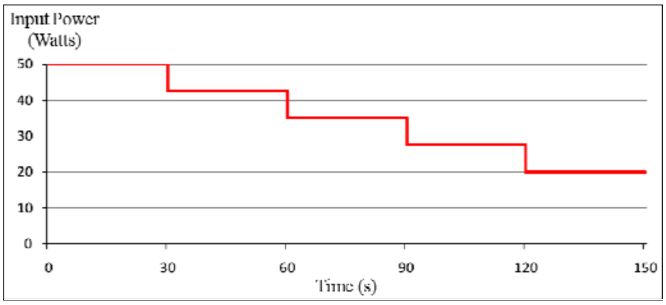


Fig. 18. Motor input power pattern for step power change.

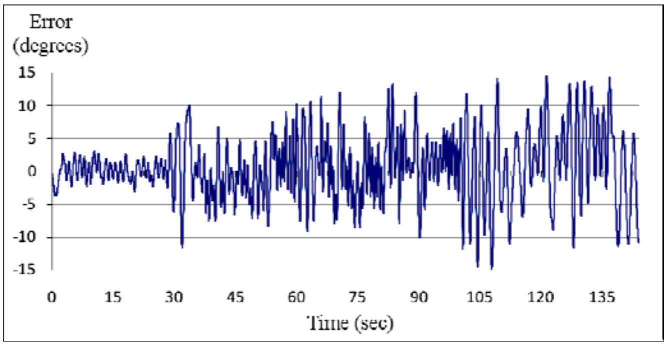


Fig. 19. Error in body-angle using generic PID controller for step power change.

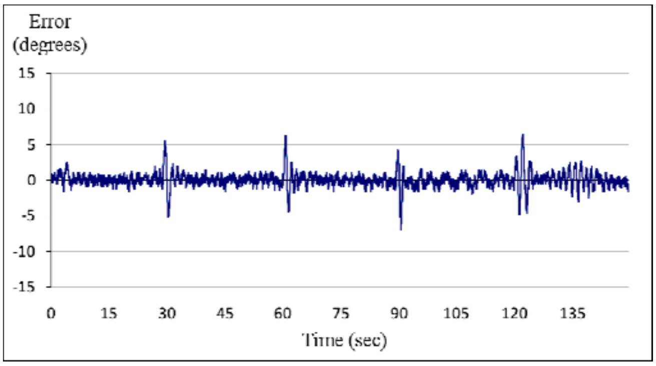


Fig. 20. Error in body-angle using the gain-scheduled PID controller for step power change.

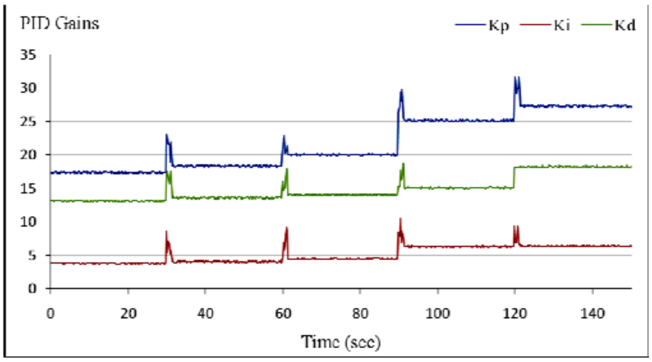


Fig. 21. Dynamic variation in PID gains using the gain-scheduled PID controller for step power change.

Table 3. Summary of performance comparison.

Power Change	PID Controller	MAE (^{o})	RMSE (^{o})	T_{set} (s)
Continuous	Fixed	14.8	6.8	-
	Adaptive	4.7	1.5	-
Step	Fixed	14.8	9.1	Does not settle
	Adaptive	6.8	2.7	1.3

The proposed adaptive PID controller achieves considerably smaller MAE and RMSE values as compared to the fixed-gain PID controller. The adaptive PID controller exhibits a faster error convergence rate and the response settles quickly within ±2.0^o of the reference after the introduction of step-change in power. The results demonstrate that the ANFIS is effective in adaptively updating the PID controller gains and maintaining desired performance under the influence of both battery depletion scenarios which could be experienced in real-life. The comparative assessment justifies the superior robustness of the proposed control strategy in rejecting the exogenous disturbances in the form of battery depletion.

8. CONCLUSION

This manuscript presents a methodology to adaptively modify PID controller gains of a TWSBR, as function of available power level, to maintain its upright posture within desired performance specifications. The need for such a methodology emanates from the proliferation of battery powered devices and the possible detrimental effects that power drainage might have on system performance. An online gain adjustment mechanism that dynamically adjusts the PID controller gains was proposed, developed and evaluated on an inherently unstable dynamic system. The proposed technique utilizes GA to acquire reference data from real-time experiments to train the ANFIS that adaptively adjusts the PID gains of the actual closed-loop system. The performance of the adaptive PID controller was compared with a fixed-gain PID controller in terms of maintaining the vertical stability of a TWSBR under two unique power-depletion scenarios. The proposed adaptive controller was able to successfully maintain the desired system performance under both power depletion scenarios, where as the fixed-gain controller exhibited a degraded posture stabilization performance. The experimental comparison validates the enhanced immunity of the adaptively controlled robotic system against the influences of uniformly or abruptly depleting battery power levels. The digital implementation of proposed controller requires single time usage of GA to evaluate the suitable gains for various power levels. The ANFIS is then used online to schedule the gains of the system which is not computationally exhaustive. In future, the proposed technique can be further investigated by applying it to other mechatronic systems. Other analytical and numerical inverse-problem approaches can also be investigated to develop similar online gain adapters.

REFERENCES

Astrom, K.J., and Hagglund, T. (2016). *Advanced PID Control*. Instrumentation, Systems, and Automation Society, North Carolina, USA.

Berenz, V., and Suzuki, K. (2011). Risk and gain battery management for self-docking mobile robots. In: *Proceedings of IEEE International Conference on Robotics and Biomimetics*, Karon Beach, Phuket, Thailand, pp. 1766 - 1771.

Bhatti, O.S., Mehmood-ul-Hasan, K., and Imtiaz, M.A. (2015). Attitude Control and Stabilization of a Two-Wheeled Self-Balancing Robot. *Control Engineering and Applied Informatics*, 17(3), pp. 98-104.

Bhatti, O.S., Tariq, O.B., Manzar, A., Khan, O.A. (2018). Adaptive intelligent cascade control of a ball-riding robot for optimal balancing and station-keeping. *Advanced Robotics*, 32(2), pp. 63–76.

Cărbureanu, M. (2014). The Development of a Neuro-Fuzzy Expert System for Wastewater pH Control. *Control Engineering and Applied Informatics*, 16(4), pp. 30-41.

Chen, Z., Xiong, R., Wang, K., Jiao, B. (2015). Optimal energy management strategy of a plug-in hybrid electric vehicle based on a particle swarm optimization algorithm, *Energies*, 8(5), pp. 3661-3678.

Chopra, V., Singla, S.K., and Dewan, L. (2014). Comparative analysis of tuning a PID controller using intelligent methods. *ACTA Polytechnica Hungarica*, 11(8), pp. 235-249.

Dalal, M., Ma, J., and He, D. (2011). Lithium-ion battery life prognostic health management system using particle filtering framework. *Proceedings of the Institution of Mechanical Engineers, Part O: Journal of Risk Reliability*, 225(1), pp. 81-90.

Elbori, A., Turan, M., and Arikan, K. B. (2018). Evaluation and Optimization of Nonlinear Central Pattern Generators for Robotic Locomotion. *Control Engineering and Applied Informatics*, 20(3), pp. 89-98.

Gui, P., Tang, L., and Mukhopadhyay, S. (2015). MEMS based IMU for tilting measurement: Comparison of complementary and Kalman filter based data fusion. In: *Proceedings of IEEE 10th Conference on Industrial Electronics and Applications*, Auckland, New Zealand, pp. 2004-2009.

Gurban, E. H., Dragomir, T. L., and Andreescu, G. D. (2014). Greenhouse Climate Control Enhancement by Using Genetic Algorithms. *Control Engineering and Applied Informatics*, 16(3), pp. 35-45.

Halal, F., and Dumitrache, I. (2006). Genetic Algorithm in Mobile Robot Control. *Control Engineering and Applied Informatics*, 8(2), pp. 21-30.

Hussain, K.M., Zepherin, R.A.R., and Shantha, M. (2014). Comparison of PID controller tuning methods with genetic algorithm for FOPTD system. *International Journal of Engineering Research and Applications*, 4(2), pp. 308-314.

Ioanaş, G. L. (2012). Modeling, Identification and Prediction of Inherent quasi-stationary Pressure Dynamics of a Common-Rail System using Neuro-Fuzzy Structures with Local Linear ARX models. *Control Engineering and Applied Informatics*, 14(3), pp. 61-70.

Jaen-Cuellar, A.Y., Romero-Troncoso, R.J., Morales-Velazquez, L., and Osornio-Rios, R.A. (2013). PID-controller tuning optimization with genetic algorithms in servo systems. *International Journal of Advanced Robotic Systems*, 10(9), pp. 1-12.

Jamaluddin, A., Sihombing, L., Supriyanto, A., Purwanto, A., and Nizam, M. (2013). Design real time battery monitoring system using LabVIEW interface for Arduino (LIFA). In: *Proceedings of 2013 Joint International Conference on Rural Information & Communication Technology and Electric-Vehicle Technology*, Bandung, Indonesia, pp. 1-4.

Kim, B.G., Patel, D.D., and Salameh, Z.M. (2013). Circuit Model of 100 Ah Lithium Polymer Battery Cell. *Journal of Power and Energy Engineering*, 1, pp. 1-8.

Kumar, V., Gaur, P., and Mittal, A. P. (2015). Novel AI based On-Line Sequential Learning Technique for High Performance DC Servo motor Control. *Control Engineering and Applied Informatics*, 17(2), pp. 3-11.

Li, Z., Yang, C., Fan, L. (2013). *Advanced Control of Wheeled Inverted Pendulum Systems*. Springer-Verlag, London, UK.

Liu, W., and Dai, J. (2015). Design of Attitude Sensor Acquisition System Based on STM32. In: *Proceedings of Fifth International Conference on Instrumentation and Measurement, Computer, Communication and Control*, Qinhuangdao, China, pp. 1850-1853.

Malysz, P., Sirouspour, S., and Emadi, A. (2014). An optimal energy storage control strategy for grid-connected microgrids. *IEEE Transactions on Smart Grids*, 5(4), pp. 1785-1796.

Mohammed, N.F., Song, E., Ma, X., Hayat, Q. (2014). Tuning of PID controller of synchronous generators using genetic algorithm. In: *Proceedings of IEEE International Conference on Mechatronics and Automation*, Tianjin, China, pp. 1544-1548.

Mosaad, M.I., Salem, F. (2014). LFC based adaptive PID controller using ANN and ANFIS techniques. *Journal of Electrical Systems and Information Technology*, 1(3), pp. 212-222.

Panday, A., and Bansal, H.O. (2014). A review of optimal energy management strategies for hybrid electric vehicle. *International Journal of Vehicular Technology*, 2014, pp. 1-19.

Petcut, F. V., and Dragomir, T. L. (2010). Solar Cell Parameter Identification using Genetic Algorithms. *Control Engineering and Applied Informatics*, 12(1), pp. 30-37.

Pourmousaviv, S.A., Nehrir, M.H., Colson, C.M., and Wang, C. (2010). Real-time energy management of a stand-alone hybrid wind-microturbine energy system using particle swarm optimization. *IEEE Transactions on Sustainable Energy*, 1(3), pp. 193-201.

Rashid, M.H. (2011). *Power Electronics Handbook*. Elsevier, Amsterdam, Netherlands.

Saha, B., and Goebel, K. (2009). Modeling Li-ion battery capacity depletion in a particle filtering framework. In: *Proceedings of the Annual Conference of the Prognostics and Health Management Society*, San Diego, CA, USA, pp. 1-10.

Saleem, O., Abbas, F., Khan, M. U., Imtiaz, M. A., and Khalid, S. (2018). Adaptive Collaborative Position Control of a Tendon-Driven Robotic Finger. *Control Engineering and Applied Informatics*, 20(2), pp. 87-99.

Serrao, L., Onori, S., Sciarretta, A., Guezennec, Y., and Rizzoni, G. (2011). Optimal energy management of hybrid electric vehicles including battery aging. In: *Proceedings of the 2011 American Control Conference*, San Francisco, CA, USA, pp. 2125-2130.

Stan, S. D., Maties, V., and Balan, R. (2007). Multi-Objective Design Optimisation of a Planar Micro Parallel Robot using Genetic Algorithms. *Control Engineering and Applied Informatics*, 9(1), pp. 41-46.

Szymak, P. (2016). Using Neuro-Evolutionary-Fuzzy Method to Control a Swarm of Unmanned Underwater Vehicles. *Control Engineering and Applied Informatics*, 18(3), pp. 82-92.

Tam, J.H., Ong, Z.C., Ismail, Z., Ang, B.C., Khoo, S.Y., and Li, W.L. (2018). Inverse identification of elastic properties of composite materials using hybrid GA-ACO-PSO algorithm. *Inverse Problems in Science and Engineering*, 26(10), pp. 1432-1463.

Toufouti, R., Meziane, S., and Benalla, H. (2009). New Direct Torque Neuro-Fuzzy Control Based SVM for Dual Two Level Inverter-Fed Induction Motor. *Control Engineering and Applied Informatics*, 11(2), pp. 3-13.

Vijayakumar, K., and Manigandan, T. (2016). Nonlinear PID Controller Parameter Optimization using Enhanced Genetic Algorithm for Nonlinear Control System. *Control Engineering and Applied Informatics*, 18(2), pp. 3-10.

Wirasingha, S.G., and Emadi, A. (2011). Classification and review of control strategies for plug-in hybrid electric vehicles. *IEEE Transactions on Vehicular Technology*, 60(1), pp. 111-122.

Yusoff, T.A.F.K., Atan, M.F., Rahman, N.A., Salleh, S.F., and Wahab, N.A. (2015). Optimization of PID tuning using genetic algorithm. *Journal of Applied Science and Process Engineering*, 2(2), pp. 97-106.

Zhou, T., and Sun, W. (2014). Optimization of battery–supercapacitor hybrid energy storage station in wind/solar generation system. *IEEE Transactions on Sustainable Energy*, 5(2), 408-415.