Brain Emotional Learning and Adaptive Model Predictive Controller for Induction Motor Drive: A New Cascaded Vector Control Topology

Muhammad Affan and Riaz Uddin*

Abstract: With the development of high-speed microprocessors, it is now possible to implement mathematically complex vector control algorithms without compromising on the performance of motor drive. Among vector control techniques space vector proportional-integral (PI), direct-torque control (DTC), field-oriented control (FOC), model-predictive control (MPC) are being widely used in industries. But their limitations have urged researchers to develop more advance techniques. In this paper, a new technique learning and adaptive model - based predictive control (termed as LAMPC) is proposed for the vector control of three phase induction motor. In the proposed method, the dynamic model of induction motor is updated adaptively based on prediction (receding horizon principle) for the inner control loop (current control) while the brain emotional learning-based intelligent controller (BELIC) is used for the outer control loop (speed control). The proposed methodology offers desired dynamic response, precise tracking, good disturbance handling capability along with satisfactory steady-state performance. To show the effectiveness of the proposed approach, benchmark simulation results for various inputs are presented using MATLAB/Simulink. Finally, the detailed qualitative and quantitative comparison of the proposed LAMPC is made with the most relevant vector techniques to show its significance.

Keywords: Induction motor, model predictive control, vector control, voltage vectors.

NOMENCLATURE

- $\Psi_s, \Psi_r$: Stator and rotor flux (space vector representation)
- $\theta_e, \theta_s$: Electrical and synchronous angle
- $\omega_e, \omega_s, \omega_r$: Electrical, slip and rotor angular speed
- $V_{sd}, V_{sq}$: ‘d’ and ‘q’ component of stator voltage in dq-frame
- $i_{sd}, i_{sq}$: ‘d’ and ‘q’ component of rotor current in dq-frame
- $\Psi_{rd}, \Psi_{rq}$: ‘d’ and ‘q’ component of rotor flux in dq-frame
- $\Psi_{r\alpha}, \Psi_{r\beta}$: ‘$\alpha$’ and ‘$\beta$’ component of rotor flux in $\alpha\beta$-frame
- $T_e$: Electromagnetic torque
- $K_{osp}, K_{osi}$: Outer speed loop proportional and integral gain
- $K_{ifp}, K_{ifi}$: Inner loop flux control proportional and integral gain
- $K_{itp}, K_{iti}$: Inner loop torque control proportional and integral gain
- $k_q, k_T$: Weighting factor for stator flux and torque
- $\alpha, \beta$: Amygdala and orbitofrontal cortex learning rate constant
- $p, q$: Weighing co-efficient for state and actuation matrices
- $T_s$: Sampling time

1. INTRODUCTION

Induction motors are among the most widely utilized electric motors in industries due to their robustness, better efficiency, and lower maintenance. However, the control of induction motor is relatively complex task because of its nonlinear torque speed curve. In general, the control strategies of induction motors are broadly classified as scalar control and vector control [1]. Scalar control uses a simple and more direct approach for controlling the speed of induction motor. However, these techniques cannot be used with systems having dynamic behavior. The scalar techniques of induction motor control are variable supply voltage control, variable rotor resistance, and constant voltage/Hz control [1]. The scalar methods have fol-
lowing inherent flaws. Firstly, the equivalent model applied with three-phase sinusoidal wave is based on steady state analysis. This could induce the high spikes of voltage and current in the system during the transient period, which could potentially reduce the overall power conversion efficiency [1]. Moreover, the control design should compensate these additional transients by increasing the size of power components [2]. Secondly, the control of three-phase circuit looks like three separate single-phase circuits, instead of single three-phase circuit. Therefore, the three-phase sinusoidal reference variables are difficult to control by PI controllers because they would distort the sinusoidal reference [3]. Thirdly, the torque and speed are interdependent on each other [4] and lastly, the poor dynamic performance is observed [5].

On the other hand, vector control techniques [1] overcome the above shortcomings of classical drives by incorporating some advanced mathematical techniques. The control of induction motor could be modeled as that of DC machine by transformation of three phase time variant into two phase time invariant system by mathematical transformations like Clark (conversion of balanced three phase quantities into time variant and balanced two phase quantities) and Park (balanced two phase orthogonal system into orthogonal rotating reference frame) [6]. The precise control of machine in both steady state and transient regions could improve the dynamic performance of induction machine [7]. The widely used field-oriented control (FOC) technique is precisely based on this concept [2] and therefore, offers better performance during start and steady-state operation but at the expense of computational cost [8]. Lower robustness is another main issue associated with the FOC’s mathematical framework [9]. In comparison, direct-torque control (DTC) is other established technique which is known for its simple structure and fast dynamic response [10] and therefore, highly suitable for lower performance applications [8]. The major limitations of DTC are sluggish response (during start and steady state) and lower steady-state performance. However, improved algorithms have been developed in the control arena such as DTC-SVM, Modified Direct Torque Control and 12- sector DTC to improve the performance of DTC [11]. Similarly, PI-control based designs for vector control applications [12] are also developed but those are relatively simple and possess many limitations ranging from static gains, parametric dependency, linearity and associated trade-offs [13]. Alternatively, many non-linear control-based approaches are being extended towards the induction motor control to address these limitations namely, but not limited to the backstepping control [14], sliding mode control [15], finite control set - model predictive control (FCS-MPC) [16]. Usually, model-driven techniques, for instance, FCS-MPC involves estimation of the unmeasured variables using measured variables, prediction of the future outputs of the system using its model, optimization of the control signal using cost function and lastly, the inverter switching (applying the optimal voltage vector to induction motor). Therefore, the modified techniques enhances the results and performance by addressing the limitations or shortcomings of these parts, but also increases the complexity of algorithm, model dependency, sensitivity and computational cost [17].

Due to the recent advancements in artificial intelligence (AI), model-driven techniques [18] are being extensively replaced by rule or data-based approaches [19], and these approaches (in which intelligence is not explicitly provided to the system, but acquired over time) have proved to be more effective [20]. In these AI or evolutionary computing-based approaches, critic (AI agent) constantly evaluates the outcome of actuations based on overall performance which in turn enables controller to learn online and gain required intelligence. Brain emotional learning-based intelligent controller (BELIC) was one such approach that adopted the computational model for mimicking the emotion producing brain’s parts such as sensory cortex, amygdala, thalamus and orbitofrontal cortex [21]. BELIC can be categorized into direct or indirect approach based on their usage [22]. In indirect approach, BELIC tunes the parameter of the required controller whereas, direct method of BELIC itself acts as a controller. This emotion based reinforcement learning approach is also being utilized in decision support systems, control engineering problems, robotics and obtaining refined results [21,23]. However, model-free approaches require intensive calculations and therefore, infeasible for practical high-speed switching operations [24].

Recently, enhanced cascaded approaches are being explored for induction motor control. In [25], the authors have reported significant improvement in performance through cascaded GPIO-PCC approach over conventional PI-PCC. The relative improvement is due to the inclusion of disturbance observer-based PI controller in outer loop which allows precise computation of reference current values which in turn allows inner predictive current control loop to yield better results. The proposed work has also taken cascaded approach but on contrary, the proposed method combines the benefit of both model-driven [26] and model-free [19] methodologies through forming a hybrid of direct BELIC and adaptive MPC for the vector control problem of Induction motor. Specifically, the outer-loop of proposed control methodology is model-free in nature, therefore, incorporates the high auto-learning rate, simple structure and independency from motor or other parameters. However, unlike [19] which also uses BELIC in inner current control loop, the proposed methodology utilizes adaptive MPC in inner loop. This cascaded structure is selected mainly due to two reasons: 1) BELIC computational complexity makes it infeasible to use in high-speed current control loop, there-
fore, as the outer-loop bandwidth is much lower than that of the internal-loop, BELIC is only used in outer-loop of the proposed methodology. 2) For inner loop current control, predictive control methodologies are well-established techniques due to their inherent stability, higher flexibility and relatively better performances especially for motor control applications [1,27]. Therefore, this proposed architecture fundamentally utilizes MPC in current control loop but augment it with features such as adaptation, skipping model update, partial pre-computation and varying receding horizon as per given the vector control problem of induction motor (discussed in Section 3). Moreover, the performance of proposed controller is compared with the conventional schemes through simulation results benchmarked with specified criteria to further validate the proposed methodology presented.

The remaining part of the paper is organized as follows: Section 2 explains the structure of the proposed LAMPC. Section 3 presents the comprehensive benchmarked simulation results of the proposed approach along with the discussion and comparison with the other relevant approaches (PI-SVPWM, FOC, DTC and FCS-MPC). Finally, Section 4 concludes the paper.

In order to maintain neutrality in comparison, bacterial foraging algorithm (BFA) is used for auto-gain tuning of all control techniques. BFA is an evolutionary computing method inspired by the foraging nature of bacteria [28]. The detail description of gain optimization process is presented in Section 3.2.

2. PROPOSED METHODOLOGY

The proposed methodology (LAMPC) is described in two parts namely the outer speed control loop and the inner current control loop.

2.1. Speed control loop

The proposed work has adopted a computational model of BELIC [29]. The inputs to the controller are the rotor speed ($\omega_r$), reference speed ($\omega_{ref}$), rotor flux ($\Psi_r$) and refererce fluxes ($\Psi_{ref}$) while $i_{sd}^{ref}$ and $i_{sq}^{ref}$ are the output signals, as shown in Fig. 1. The inputs to the outer speed loop ($\omega_r$, $\omega_{ref}$, $\Psi_r$ and $\Psi_{ref}$) are transformed through stimuli generator function into Sensory Inputs (SI) which is then feed into amygdala and orbitofrontal cortex. The output signal $i_{sd}^{ref}$ is obtained by subtracting orbitofrontal cortex’s output from amygdala’s output as shown in Fig. 1 and (1). The subtraction of large orbitofrontal cortex output ($W_{ad}^sdSI_{ad}$) in (1) depicts the unlearning characteristic of amygdala [22,30].

$$i_{sd}^{ref} = V_{sd}^T SI_{ad} - W_{ad}^T SI_{ad}, \quad (1)$$

where $SI_{ad}$ is the sensory signal derived from inputs ($\omega_r$ and $\omega_{ref}$) as shown in (2). The $N()$ function in (2) is the unity-based normalization mathematically expressed in (3). The $V_{sd}$ and $W_{ad}$ are amygdala and orbitofrontal cortex adaptive weights respectively for sensory signal $SI_{ad}$. The adaptations to $V_{sd}$ and $W_{ad}$ weights are accomplished through (4) and (5) [22,30].

$$SI_{ad} = N \left( \begin{bmatrix} \omega_{ref} \\ e_{t0} \end{bmatrix} \right), \quad (2)$$

$$X' = \frac{X - X_{min}}{X_{max} - X_{min}}, \quad (3)$$

$$V_{sd} = V_{sd}^{old} + \alpha \max \left( 0, SI_{ad} \left[ (K_{osf}e_{t0} + K_{osi} \int e_{t0} dt) \right], \quad (4) \right.$$  

$$W_{ad} = W_{ad}^{old} + \beta SI_{ad} \left[ i_{sd}^{ref} - (K_{osf}e_{t0} + K_{osi} \int e_{t0} dt) \right]. \quad (5)$$

The max function in (4) implies a permanent learning characteristic of amygdala i.e., amygdala is incapable to unlearn any emotional response learned overtime. This biological incapability is modelled through non-decreasing mathematical form in (4) i.e., the non-negative increment causes $V_{sd}$ to either increase or remain same.

The orbitofrontal cortex gain update $W_{sq}$ in (5) allows both positive and negative increments. It also needs to be noted here that the adaptations in (4) and (5) can also be externally governed by the user defined proportional and integral terms. These terms act as Emotional Cues (EC) generated through reward generator based on error values and passed onto amygdala and orbitofrontal cortex. These emotional cues influence the incrementation in $V_{sd}$ and $W_{ad}$ weights thereby, reinforcing or suppressing $i_{ad}^{ref}$.

The output signal $i_{sq}^{ref}$ can also be obtained in a similar fashion as (1) being shown below.

$$i_{sq}^{ref} = V_{sq}^T SI_{sq} - W_{sq}^T SI_{sq} \quad (6)$$
Here, $S_I$ is the sensory signal derived from inputs ($\Psi$ and $\Psi_{ref}$) as shown in (7). Likewise, $V_s$ and $W_s$ are amygdala and orbitofrontal cortex adaptive weights respectively for sensory signal $S_I$ shown in (8) and (9).

$$S_I = N \left( \begin{bmatrix} \Psi_{ref} \\ e_y \end{bmatrix} \right),$$

$$V_s = V_s^{old} + \alpha \max \left( 0, S_I \right) \left[ \begin{bmatrix} (e_y + \int e_y dt) \\ -(V_s^{old} + S_I) \end{bmatrix} \right],$$

$$W_s = W_s^{old} + \beta S_I \left[ \begin{bmatrix} e_y \\ \int e_y dt \end{bmatrix} \right].$$

The above mathematical formulation of BELIC based outer speed loop control methodology is summarized as block diagram in Fig. 1 and as pseudo-code in below.

**Initialization:** Set $V_{sd}$, $V_{sq}$, $W_{sd}$, $W_{sq}$ = 0

**Define** ECs ($K_{ossp}$, $K_{osi}$ and objective function)

**for** each iteration $t = t_s$ **do**

Compute $S_I_{sd}$ and $S_I_{sq}

Compute $I_{sd}^{ref}$ and $I_{sq}^{ref}

Update $V_{sd}$ and $V_{sq}$

Update $W_{sd}$ and $W_{sq}$

**end for**

### 2.2. Current control loop

For current control loop (unlike existent step-ahead FCS-MPC technique), the proposed technique computes optimal voltage vector based on receding horizon principle. As the MPC performance is model dependent and induction motor model is highly non-linear in nature, there always exists a trade-off between performance and computational burden. Therefore, this paper presents a way of dealing with non-linearity associated with the model through successive linearization at each control instant and then feeding it into the adaptive MPC. Thus, this provides better performance as evident from results, and is also less computationally expensive in comparison to conventional MPC. Moreover, the proposed technique ingeniously incorporates few additional features to resolve excessive computation problem as discussed in the following paragraphs. The proposed technique along with previously discussed BELIC block is shown in Fig. 2 to supplement theoretical details. A pseudocode of inner control loop is also provided in the end of this section to assist readers’ comprehension.

The state, input and output variables are defined for each interval $t$ as $x(t) \in R^r$, $u(t) \in R^p$ and $y(t) \in R^q$, respectively, as given in (10) as shown below.

$$\dot{x} = Ax + Bu,$$

where $x = [i_{sd} \ i_{sq}]^T$, $u = [V_{sd} \ V_{sq}]^T$ while $A$, $B$ and $C$ are state-space matrices defined as below.

$$A = \left[ \begin{array}{cc} -\frac{1}{\tau_s} & \omega_c \\ -\omega_c - \frac{1}{\tau_{is}} & \frac{1}{\tau_{is}} \end{array} \right], \quad B = \left[ \begin{array}{c} \frac{1}{\tau_{is}} \\ 0 \end{array} \right].$$

The above state-space matrices are carefully selected such that least computation is required at each interval. As evident, only $A$ is time varying in nature, creating time dependency due to the instantaneous values of the slip speed $\omega_s$, i.e., $A = A(\omega_s(t))$, restricting an offline computation of equivalent system model. Thus, this requires a time-varying solution of (10) to be updated at each time instant with the new values of $\omega_s$. However, $\omega_s$ variations are almost insignificant during steady-steady as evident from (11), therefore proposed methodology skips model update during steady-state period, thereby, lessening the computational expense.

$$\omega_s = \omega_s + \frac{1}{\tau_{is}} i_{sq},$$

The time dependency along with the associated computational burden of MPC poses a serious limitation in comparison to existing motor control techniques, the proposed technique eases the computational burden by skipping model update (as discussed above), partial pre-computing [31] and varying horizons [32]. The idea of pre-computation of system matrices was originally used in FCS-MPC [31], but as the MPC computational burden is relatively significant in comparison to FCS-MPC, the partial pre-computation in proposed technique is more effective. Moreover, MPC computational complexity and performance are directly associated with the lengths of prediction ($p$) and control ($m$) horizons [33]. Therefore, performance and stability can be enhanced by increasing $p$ and $m$ which in turn increase memory requirement, solution and computation time [26,33]. To ensure faster computation without degrading performance, length $p$ is shrunk during steady-state by the proposed technique.
It is worth restating, system (10) requires discrete-time state-space conversion for implementation by digital microprocessor. The establish methods for doing so are direct calculation, Euler and modified Euler methods. Euler is simple yet less accurate, whereas others are more accurate but less simple.

The following constrained optimization problem is computed by MPC at each time interval $r$.

$$
\min_{\{u_0, \ldots, u_{n-1}\}} \left\{ \left( x_{t+p} - x_{t+p}^{ref} \right)^T P \left( x_{t+p} - x_{t+p}^{ref} \right) \right. \\
+ \sum_{k=0}^{p-1} \left[ \left( x_{t+k} - x_{t+k}^{ref} \right)^T Q \left( x_{t+k} - x_{t+k}^{ref} \right) + u_{t+k}^T R u_{t+k} \right],
\right. \right.
\tag{12}
$$

where

$$
x_{\text{min}} \leq x_{t+k} \leq x_{\text{max}}, \quad k = 1, \ldots, p - 1,
$$

$$
u_{\text{min}} \leq u_{t+k} \leq u_{\text{max}}, \quad k = 1, \ldots, p - 1,
$$

where $u_0, \ldots, u_{t+k-1}$ is the input sequence applied to model (6) for predicting state vector $x_{t+k}$ i.e., till time instant $t + k$ starting from the state $x_t$. Here, $m$ and $p$ represent moving and prediction horizons, respectively. The $Q$ and $R$ terms represent the state and input weighing matrices with respective dimensions, and assumed to hold $R = R^T > 0$, $Q = Q^T \geq 0$. These $Q$ and $R$ matrices include $q$ and $r$ terms as the diagonal weighing elements, whereas $P$ is obtained by solving the algebraic Riccati equation as shown in

$$
P = A^T P A - A^T (B^T P B + R)^{-1} B^T P A + Q.
\tag{13}
$$

Several techniques are presented in [26,27,34] to solve (12) online optimization problem. However, such techniques are not discussed for brevity and only concise mathematical formulation is elaborated. Initially, the state vector ($\in \mathbb{R}^{np}$) equation for prediction horizon $p$ is obtained as shown in

$$
X = \Lambda x_t + \Theta U,
\tag{14}
$$

where

$$
\Lambda = \begin{bmatrix} A & A^2 & \cdots & A^{p-1}B \end{bmatrix}, \quad \Theta = \begin{bmatrix} B & 0 & \cdots & 0 \\ AB & B & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ A^{p-1}B & A^{p-2}B & \cdots & B \end{bmatrix},
$$

$$
X = \begin{bmatrix} x(t+1|t) \\ x(t+2|t) \\ \vdots \\ x(t+p|t) \end{bmatrix}, \quad U = \begin{bmatrix} u(t|t) \\ u(t+1|t) \\ \vdots \\ u(t+p-1|t) \end{bmatrix}.
$$

Now, (15) is substituted into (12) yielding the simplified finite horizon optimal control equation for minimization shown in

$$
\min_U \{ U^T H U + 2 f U + J_o \} \quad \text{(s.t. } aU \leq b),
\tag{15}
$$

where

$$
H = \Theta^T Q \Theta + R,
$$

$$
f = \Theta^T Q (\Lambda x_{t+p+1|t} - x_{t+p+1|t}^{ref}),
$$

$$
J_o = (\Lambda x_{t+p+1|t} - x_{t+p+1|t}^{ref})^T Q (\Lambda x_{t+p+1|t} - x_{t+p+1|t}^{ref}),
$$

$$
\tilde{Q} = \begin{bmatrix} Q & 0 \\ 0 & P \end{bmatrix}, \quad \alpha = \begin{bmatrix} -\Theta \\ I_{np} \end{bmatrix}, \quad \beta = \begin{bmatrix} X_{\text{max}} - 2\alpha(k) \\ -I_{np} \end{bmatrix}.
$$

The $X_{\text{max}}, X_{\text{min}}$ comprised of $p$ vectors of upper and lower limits of stator current as per (16). Likewise, the constraints on the amplitude of control signal (17) are formulated as $U_{\text{min}}, U_{\text{max}}$ containing $p$ applied stator voltage vectors with upper and lower limit ($I_{np}$ and $-I_{np}$, respectively).

The goal is to minimize (15), derived from linear state space model, subjected to constraints. The optimization problem (15) is quadratic in nature subject to linear inequality constraints thus, the problem could be uniformly solved by quadratic programming. Numerical solution methods of quadratic programming such as Interior-point, Active-set and Trust-Region methods are equally applicable to the given optimization problem. However, an active-set approach is chosen in this work primarily due to its higher reliability, robustness and data-efficiency [35]. Active-set method decides active set of constraints based on KKT conditions [36]. The MATLAB quadprog toolbox is being utilized as solver for quadratic optimization. For theoretical background and detailed insight into the implementation of MPC, refer the following [26,34].

**Initialization:** Define system’s model, constants and coefficients
1) Acquire $i_{sd}^{ref}, i_{sq}^{ref}$ and $\omega_m$
2) Estimate $\omega_m, i_{sd}$ and $i_{sq}$
3) **function** Adaptive_MPC_Algorithm (inputs

- state-space matrices, initial control variable matrix, initial manipulated variable matrix, control horizon length, predictive horizon length, sampling time)
- a. Determine Control Structure
- b. Construct Prediction matrices
- c. Update model predictions
- d. Check ill-conditioning
3. SIMULATION WORK

3.1. Induction motor specifications

The induction motor with specifications used for benchmark results of discussed approaches are shown in Table 1.

For the given case of induction motor, it is subjected to current and voltage constraints. The current constraints are considered because high starting stator current values could be extremely dangerous for inverters [31] whereas, the input voltage constraints are due to the maximum providable voltage limit $V_{dc}$ [16]. The following approximation of inequality constraints are used as shown below.

$$-40 \leq i_{sd}, i_{sq} \leq 40,$$
$$-216.35 \leq v_{sd}, v_{sq} \leq 216.35.$$  \hspace{1cm} (16)  \hspace{1cm} (17)

3.2. Tuning the gain parameters

The bacterial foraging algorithm (BFA) allows cells to move stochastically and collectively towards the optimum solution (which in our case is optimal gain values). This algorithm accomplishes this objective with the help of three processes namely Chemotaxis (tumbling or swimming), reproduction and elimination-dispersal. The process is initiated by bacterium chemotaxis such as swimming and tumbling via flagella. Swimming implies directed motion while tumbling is undirected random motion by bacterium [28]. These movements enable localization of gain parameters of all discussed technique for minimization of fitness functions thereby providing BFA optimized gains [37]. The fitness evaluation is done using the Integral of the time weighted absolute error (ITAE) as shown below.

$$f(t) = \int_0^t t|e(t)|dt.$$  \hspace{1cm} (18)

Reproduction process is initiated after the chemotaxis and swarming. In this process, a portion of gain parameters having the least fitness values are replaced with reproduced population of gain parameter from high fitness values. Reproduction process is followed by the Elimination-dispersal step. In this step, a portion of gain parameter is attenuated or rejected. The parameters are attenuated randomly within the specified range while rejection is done based on poor fitness values. This information flow has been shown in Fig. 3.

Table 2 lists all the inner and outer loop tuning parameters against their respective induction motor vector control technique. BFA has been used to pre-tune all these control techniques independently i.e., their corresponding gains are tuned offline. The $\omega_r$, $\Psi_s$ and $T_e$ error values in DTC, FOC, PI-SVPWM, FCS ($\alpha\beta$) and FCS ($dq$) control techniques are used for fitness evaluation of the gain parameters.

![Fig. 3. Flowchart of BFA.](image-url)
Table 2. Discussed techniques' corresponding gains.

| Techniques    | Tuning parameters |
|---------------|-------------------|
| DTC           | $K_{isp}, K_{isi}$ |
| FOC           | $K_{isp}, K_{isi}, K_{isp}, K_{isi}$ |
| PI-SVPWM      | $K_{isp}, K_{isi}, K_{isp}, K_{isi}, K_{isp}, K_{isi}$ |
| FCS ($\alpha\beta$) | $K_{isp}, K_{isi}, K_{isp}, k_T$ |
| FCS ($dq$)    | $K_{isp}, K_{isi}, k_S, k_T$ |
| LAMPC         | $K_{isp}, K_{isi}, \alpha, \beta, q, r$ |

ters as per (18), whereas fitness evaluation in LAMPC is done through $\omega_r, \Psi_s$ and $I_s$ error values.

3.3. Benchmark results of LAMPC

The proposed LAMPC is applied on induction motor with specifications as shown in Table 1. The speed and torque response of LAMPC are shown in Fig. 4 respectively. Varying step signals at multiple time instants are applied to the induction motor having 5 N-m load torque.

External disturbance handling capability of LAMPC is depicted in Fig. 5. When an external load torque of 20 N-m is added to induction motor during its steady-state operation of 540 rpm at 0.2 sec, an equivalent but counter electromagnetic torque is developed by Induction motor. In spite of this external disturbance, the rotor speed showed almost insignificant speed drop which quickly returns to its original value whereas, stator flux remains constant. Thus, it can be concluded that complete decoupling between motor torque and stator flux is realized in LAMPC with strong robustness against external load variance.

3.4. Benchmark results of other relevant approaches with comparisons and discussions

The discussed techniques are evaluated based on the following criteria of comparison as below:

1) Performance comparison via transient and steady-state performance using step signal.
2) Analysis of total harmonic distortion (THD) in stator current signal.
3) Quantitative and qualitative assessment and characterization.

Based on the above defined criteria, all discussed techniques namely FCS-MPC ($\alpha\beta$ and $dq$), FOC, PI-SVPWM, DTC are simulated and compared against LAMPC in Figs. 6 to 8.

The first simulation was done to test the steady state and dynamic speed response of LAMPC against MPC-FCS, PI-SVPWM, FOC and DTC. From Fig. 6, it is evident that LAMPC is the quickest to reach the set-point signal under the given constraints. The other vector control techniques seem to have indifferent dynamic performance in response to the reference input speed signal. Interestingly, MPC-FCS ($\alpha\beta$ and $dq$) have same speed response which re-accentuate the fact that both methods are almost same just their implementation are different.

The second simulation results i.e., Fig. 7 compares the associated torque response of all discussed techniques with LAMPC. The proposed LAMPC method seems to have the least torque ripples among all compared techniques. Moreover, it quickly tracks external and internal torque disturbances i.e., 5 N-m and 40 N-m respectively. The high torque ripples in FCS ($\alpha\beta$) and FCS ($dq$) are due to inequality constraints.

Fig. 8 illustrates the associated flux response of all discussed techniques. The proposed LAMPC method seem to have least ripples among all simulated techniques. Moreover, it is the quickest to reach the rated 0.71 Wb. The incompetency to closely track flux along with high flux ripples in Fig. 8 could be attributed to FCS ($\alpha\beta$) and ($dq$) methodology that ought to balance between flux and torque tracking in [38] resulting in inadequate decoupling between torque and flux in these methods.

The harmonic spectrum of stator current at low speed and high-speed steady-state operation without load is shown in Fig. 9 to Fig. 14. The data of stator currents of all discussed techniques are acquired and then analyzed in MATLAB using fast Fourier analysis (FFT) for THD present in stator currents. It can be clearly seen that proposed technique i.e., LAMPC has the least current THD...
in both low-speed and high-speed operations i.e., 1.32% and 1.61%, respectively. The Table 3 summarizes as well as presents the basic qualitative comparison of transient, steady-state and THD of all discussed techniques.

As far as the textual topological assessment and comparison of discussed techniques are concerned, the theoretical basis for elevated performance of proposed controller are explained here. In this regard, the absence of current regulation loop, control realization through predefined switching table and hysteresis comparators are the key differences of DTC. This control formulation requires less computation that limits control resolution and hinders quiescent state operation i.e., steady-state performance deterioration (error fluctuates between hysteresis band as evident from Fig. 11). The adjustment in hysteresis band gap can be made to alter error tolerance and average switch-
Fig. 11. Harmonic spectrum of stator current for DTC at (a) 150 rpm (b) 1500 rpm.

Fig. 12. Harmonic spectrum of stator current for FOC at (a) 150 rpm (b) 1500 rpm.

Fig. 13. Harmonic spectrum of stator current for PI-SVPWM at (a) 150 rpm (b) 1500 rpm.

Fig. 14. Harmonic spectrum of stator current for LAMPC at (a) 150 rpm (b) 1500 rpm.
Table 3. Quantitative comparison among approaches in tabular form.

|                         | Transient response | Steady-state response | THD (%) | Computation |
|-------------------------|--------------------|-----------------------|---------|-------------|
|                         | Rise time (sec)    | Overshoot (%)         | Settling time (sec) | Steady-state error (%) | At low speed (150 rpm) | At high speed (1500 rpm) | Differential or integral equations** | Iteration time* (ms) |
| PI-SVPWM                | 0.09               | 0.2                   | 0.17     | 0.9          | 11.80               | 30.84                | 2                               | 0.31                        |
| FOC                     | 0.074              | 0                     | 0.118    | 0.2          | 4.66                | 36.61                | 5                               | 0.64                        |
| DTC                     | 0.28               | 0.2                   | 0.35     | 0.8          | 48.38               | 26.66                | 2                               | 0.29                        |
| FCS (αβ)                | 0.086              | 0                     | 0.112    | 0.02         | 3.59                | 2.63                 | 15                              | 0.87                        |
| FCS (dq)                | 0.076              | 0.06                  | 0.103    | 0.3          | 14.04               | 27.35                | 15                              | 0.81                        |
| LAMPC                   | 0.071              | 0                     | 0.096    | 0.04         | 1.32                | 1.61                 | 9                               | 3.8                         |

* Using Intel Core M-5Y10c CPU 0.8GHz with 8GB installed memory
** Refers to the number of differential or integral equations need to be solved in each iteration

Table 4. Summarized qualitative assessment in tabular form.

|                         | Reference frame | Co-ordinate conversion | Controlled variables | PWM modulator | Switching frequency | Flexibility |
|-------------------------|-----------------|------------------------|----------------------|---------------|---------------------|-------------|
| PI-SVPWM                | Rotating        | Needed                 | Torque, flux, V<sub>sq</sub>, V<sub>sd</sub> | Needed        | Consistent          | Low         |
| FOC                     | Rotating        | Needed                 | i<sub>sq</sub>, i<sub>sd</sub>, rotor flux, torque | Needed        | Consistent          | Low         |
| DTC                     | Stationary      | Not Needed             | Torque, stator flux  | Not Needed    | Varies              | Medium      |
| FCS (αβ)                | Stationary      | Not Needed             | Torque, stator flux  | Not Needed    | Varies              | Medium      |
| FCS (dq)                | Rotating        | Not Needed             | Torque, stator flux  | Not Needed    | Varies              | Medium      |
| LAMPC                   | Rotating        | Needed                 | Torque, rotor flux, isq, isd | Needed       | Consistent          | High        |

The main benefit of predictive control methodologies is their added flexibility in the form of cost function. Additional control sub-objectives can be added in FCS (αβ), FCS (dq) and LAMPC by modifying the cost function such as common-mode voltage reduction, constraints minimization, switching behavior etc. However, LAMPC provides additional flexibility than the other two in the form of model adaptation, provision of inherent constraint minimization, online horizon scheduling and auto-learning as discussed in Section 2. The above discussion is summarized in Table 3 and Table 4.

4. CONCLUSION

The new approach based on BELIC-AMPC is proposed and validated against the relevant vector control approaches with respect to reference tracking and disturbance rejection. It is quantitatively and qualitatively analyzed that the proposed LAMPC exhibits better transient and steady-state performance through adaptation to varying plant dynamics as evident from results. Moreover, the inherent constraints handling and horizon scheduling capability of LAMPC incorporates operational flexibility in design.

The future directions of the current work will be 1) the comprehensive experimental validation of proposed approach and the other approaches discussed in this work, 2) exploring the prospects of further lessening the parametric dependency and computational burden of control algorithm by incorporating Deep Reinforcement learning based model-free control methodology, 3) lastly the sensitivity analysis of motor parameters and their impact on controlling approaches similar to [39].

APPENDIX A

For a squirrel cage induction motor, the space vector voltage equation is

$$\ddot{V}_r = R_i I_r + \frac{d}{dt} \dot{\psi}_r,$$

(A.1)

where $R_i$ is the stator resistance, $I_r$ and $\dot{\psi}_r$ are the space vector of stator current and stator flux respectively with respect to its own winding system (stator reference frame). The space vector voltage equation of the rotor is

$$\ddot{V}_r^* = R_i I_r + \frac{d}{dt} \dot{\psi}_r^*.$$

(A.2)
All terms in above equation are in rotor reference frame (represented by *). For the synchronization of space vectors of both stator and rotor,
\[ \vec{V}_r = \vec{V}_r^* e^{+i\theta} = \vec{I}_r^* e^{+i\theta} \text{ and } \vec{I}_r = \vec{I}_r^* e^{+i\theta}. \]  
(A.3)

The \( \theta \) is an electrical angle. Since rotor winding is lagging the stator winding by \( \theta \), therefore, to synchronize rotor winding and stator winding the rotor winding is advanced by \( \theta \). Now, putting (A.3) in (A.2).
\[ \vec{V}_r = R_s \vec{I}_r + \frac{d}{dt} \vec{L}_r \vec{I}_r - j \omega_e \vec{I}_r. \]  
(A.4)
The stator and rotor fluxes are represented as
\[ \vec{\psi}_s = L_s \vec{I}_s + L_h \vec{I}_r, \]  
(A.5)
\[ \vec{\psi}_r = L_h \vec{I}_s + L_r \vec{I}_r. \]  
(A.6)
Substituting the value of \( \vec{\psi}_s \) from (A.5) in (A.1),
\[ \vec{V}_s = R_s \vec{I}_s + \frac{d}{dt} \left( L_s \vec{I}_s + L_h \vec{I}_r \right). \]  
(A.6)
Substituting the value of \( \vec{I}_r \) from (A.6) to above equation,
\[ \vec{V}_s = R_s \vec{I}_s + \frac{d}{dt} \left( L_s \vec{I}_s + L_h \vec{I}_r \right) + \frac{L_h}{L_r} \frac{d}{dt} \vec{\psi}_r. \]  
(A.7)
Substituting the value of \( \vec{I}_r \) from (A.6) to (A.4),
\[ \frac{d}{dt} \vec{\psi}_r = - \vec{\psi}_r R_s \frac{L_h}{L_r} + \frac{L_h}{L_r} \frac{d}{dt} \vec{I}_r + \vec{\psi}_r j \omega_e. \]  
(A.8)
Substituting (A.8) in (A.7),
\[ \vec{V}_s = R_s \vec{I}_s + L_s \frac{d}{dt} \vec{I}_s + \left( 1 - \frac{L_h^2}{L_r L_s} \right) \frac{d}{dt} \vec{I}_r + \vec{\psi}_r \frac{L_h}{L_r} \frac{d}{dt} \vec{I}_r + \vec{\psi}_r j \omega_e. \]  
(A.9)
Let
\[ X = \vec{\psi}_r \left( \frac{L_h}{L_r} \frac{d}{dt} \vec{I}_r + \vec{\psi}_r j \omega_e \right). \]  
(A.10)
Now, (A.9) becomes
\[ \vec{V}_s = R_s \vec{I}_s + L_s \frac{d}{dt} \vec{I}_s + L_r \left( 1 - \frac{L_h^2}{L_r L_s} \right) \frac{d}{dt} \vec{I}_r + X. \]  
(A.11)
Substituting the value of \( \vec{I}_r \) from (A.5) in (A.6),
\[ \vec{\psi}_r = \left( L_h - \frac{L_h L_r}{L_s} \right) \vec{I}_s + \frac{L_r}{L_h} \vec{\psi}_s. \]  
(A.12)
Substituting (A.11) in (A.9),
\[ X = \begin{bmatrix} \left( L_h - \frac{L_h L_r}{L_s} \right) \vec{I}_s \\ + \frac{L_r}{L_h} \vec{\psi}_s \end{bmatrix} \begin{bmatrix} -L_r R_s + \frac{L_h}{L_s} j \omega_e \end{bmatrix}. \]  
(A.13)
Simplifying (A.13) and putting \( L_1 \),
\[ L_1 = \frac{L_h}{L_r}, \]  
\[ X = j \omega_e L_4 \vec{I}_s - j \omega_e L_5 \vec{I}_r + j \omega_e \vec{\psi}_s, \]  
\[ -L_3^2 R_s \vec{I}_s + \frac{R_s L_r}{L_r} \vec{I}_r - \frac{R_s}{L_r} \vec{\psi}_s. \]  
(A.14)
Substituting the value of \( X \) from (A.14) in (A.11),
\[ \vec{V}_s = \begin{bmatrix} \left( R_s + \frac{R_s L_r}{L_r} \right) \vec{I}_s + j \omega_e \left( \frac{L_h^2}{L_r} - L_s L_r \right) \vec{I}_r + \left( 1 - \frac{L_h^2}{L_r L_s} \right) \frac{d}{dt} \vec{I}_r + \frac{L_r}{L_r} \vec{\psi}_r \\ + \left( L_r L_s - \frac{L_h^2}{L_r} \right) \frac{d}{dt} \vec{I}_r + \vec{\psi}_r - \frac{L_h}{L_r} \vec{I}_r + \vec{\psi}_r \end{bmatrix} \]  
(A.15)
Puting below assumption in (A.15), we get (A.16).
\[ \lambda = \frac{1}{L_s L_r - \frac{L_h^2}{L_r}}, \]  
\[ \frac{d}{dt} \vec{I}_s = \vec{V}_s \lambda + \lambda (R_s - j \omega_e L_r) \vec{\psi}_r + \left( R_s + \frac{R_s L_r}{L_r} \right) \lambda + j \omega_e \vec{\psi}_r. \]  
(A.16)
Assuming
\[ U = \vec{V}_s, \quad X_1 = \vec{I}_s, \quad X_1 = \frac{d}{dt} \vec{I}_s, \]  
\[ X_2 = \vec{\psi}_r, \quad X_2 = \frac{d}{dt} \vec{\psi}_r. \]  
(A.17)
Putting above considerations in (A.16),
\[ X_1 = U \lambda - \left( \left( R_s + \frac{R_s L_r}{L_r} \right) \lambda + j \omega_e \right) X_1 + \lambda (R_s - j \omega_e L_r) X_2. \]  
(A.18)
Putting (A.16) in (A.1), we get
\[ \dot{X}_2 = U - R_s X_1. \]  
(A.19)
Using (A.18) and (A.19), we get
\[ \begin{bmatrix} \dot{X}_1 \\ \dot{X}_2 \end{bmatrix} = \begin{bmatrix} L_r \lambda & 0 \\ 1 & 0 \end{bmatrix} \begin{bmatrix} L_r \lambda & 0 \\ 1 & 0 \end{bmatrix} \begin{bmatrix} X_1 \\ X_2 \end{bmatrix}. \]  
(A.20)
The above space vector-based state space model is not linearly time invariant. We will proceed further to achieve linearization and time-invariance at steady-state. Put below \( \tau \) assumption in (A.8) to get
\[
\tau \frac{d}{dt} \dot{\psi} + \dot{\psi} = j\omega_c \dot{\psi}_r + L_d i_d.
\] (A.21)

Put below assumptions in (A.21) to get (A.22),
\[
U = \dot{V}_r, X_1 = \dot{I}_r, X_1 = \frac{d}{dt} \dot{I}_r, X_2 = \frac{d}{dt} \dot{\psi}_r,
\]
\[
\dot{X}_2 = \left( j \frac{\omega_c \tau_r - 1}{\tau_r} \right) X_2 + \frac{L_d}{\tau_r} X_1.
\] (A.22)

From (A.9), we get
\[
\dot{V}_s = \left( -\frac{L_d R_c}{L_c^2} + \frac{L_d}{L_c} j\omega_c \right) \dot{\psi}_r + \left( \frac{R_s + R_c L_d^2}{L_c^2} \right) \dot{I}_r = L_s \left( 1 - \frac{L_d^2}{L_s L_c} \right) \frac{d}{dt} \dot{I}_r.
\] (A.23)

A more compact expression can be obtained by simplifying and replacing specific constants and coefficients in above equation as shown below.
\[
\sigma = 1 - \frac{L_d^2}{L_s L_c} \quad k_r = \frac{L_d}{L_c} \quad r_c = R_s + R_c L_d^2, \quad r_s = \frac{\sigma L_c}{r_r}.
\]

Also, to achieve more simplified and linearize formulation, we need to rotate reference axis with angular displacement of \( \theta_r \) (dq transformation) as shown below.
\[
\begin{align*}
\tilde{u}_s' &= \tilde{u}_s e^{-j\theta_r} = V_{sd} + jV_{sq}, \\
\tilde{\psi}_r' &= \tilde{\psi}_r e^{-j\theta_r} = \psi_{sd} + j\psi_{sq},
\end{align*}
\]

where \( \tilde{u}_s', \tilde{\psi}_r' \) are the stator voltage, stator current and rotor flux vectors referred to dq frame using angular rotation \( \theta_r \), \( \omega_r \) is angular speed of rotating flux. Considering aforementioned transformed vectors, (A.23) can now be rewritten as
\[
\begin{align*}
\tilde{I}_r' + \tau_r \left( \frac{d}{dt} \tilde{I}_r' + j\omega_r \tilde{I}_r' \right) &= k_r \left( \frac{1}{r_s} - j\omega_r \right) \tilde{\psi}_r' + \frac{1}{r_s} \tilde{u}_s'.
\end{align*}
\] (A.24)

The imaginary and real components of (A.24) are distributed and differentiated in dq-components, the following equations are obtained:
\[
\begin{align*}
\frac{d}{dt} i_{sd} &= \frac{1}{\tau_s} i_{sd} \\
&+ \omega_c i_{sq} + \frac{k_r}{r_s \tau_s} \psi_{sd} + \frac{1}{r_s \tau_s} V_{sd},
\end{align*}
\] (A.25)
\[
\frac{d}{dt} i_{sq} = - \omega_c i_{sd}
\]
\[
\begin{align*}
- \frac{1}{\tau_r} i_{rd} - \frac{k_r}{r_c \tau_r} \omega_c \psi_{rd} + \frac{1}{r_c \tau_r} V_{sq}.
\end{align*}
\] (A.26)

Since, the rotor flux is fixed to real axis. Therefore, the rotor flux \( \psi_r \) is assumed zero. Now, consider following state space formulation:
\[
\begin{bmatrix}
X_1 \\
X_2
\end{bmatrix} =
\begin{bmatrix}
-\frac{1}{\tau_r} & \omega_c & 0 & 0 \\
0 & -\omega_c & 1 & 0
\end{bmatrix}
\begin{bmatrix}
\dot{X}_1 \\
\dot{X}_2
\end{bmatrix} +
\begin{bmatrix}
\frac{1}{\tau_s \tau_r} & 0 \\
0 & \frac{1}{\tau_r \tau_s}
\end{bmatrix}
U.
\] (A.27)

REFERENCES

[1] B. K. Bose, *Modern Power Electronics and AC Drives*, Prentice Hall PTR, 2002.

[2] Texas Instruments, “Field oriented control of 3-phase AC-motors,” Literature number BPRA073, 1998.

[3] S. Z. Abbas, “Simulation, implementation and testing of three-phase controlled power inverter behavior,” Escola Tècnica Superior d’Enginyeria Industrial de Barcelona Simulation, 2016.

[4] P. K. Behera, M. K. Behera, and A. K. Sahoo, “Comparative analysis of scalar & vector control of Induction motor through modeling & simulation,” *Int. J. Innov. Res. Electr. Electron. Instrum. Control Eng.*, vol. 2, no. 4, pp. 2321-2004, 2014.

[5] ABB, “Technical guide no. 1 direct torque control - the world’s most advanced AC drive technology,” 2011.

[6] V. Gopal, “Comparison between direct and indirect field oriented control of induction motor,” *Int. J. Eng. Trends Technol.*, vol. 43, no. 6, pp. 364-369, 2017.

[7] M. Revathi and C. M. Raj, “Improved steady state and transient state torque / speed response of induction motor using fuzzy logic controller,” *Imperial Journal of Interdisciplinary Research*, vol. 2, no. 5, pp. 1697-1703, 2016.

[8] F. Wang, Z. Zhang, X. Mei, J. Rodríguez, and R. Kennel, “Advanced control strategies of induction machine: Field oriented control, direct torque control and model predictive control,” *Energies*, vol. 11, no. 1, 120, 2018.

[9] L. Bascetta, G. Magnani, P. Rocco, and A. M. Zanchettin, “Performance limitations in field-oriented control for asynchronous machines with low resolution position sensing,” *IEEE Trans. Control Syst. Technol.*, vol. 18, no. 3, pp. 559-573, 2010.

[10] X. Ji, D. He, and Y. Ge, “Study of direct torque control scheme for induction motor based on torque angle closed-loop control,” *Open Electr. Electron. Eng. J.*, vol. 9, no. 1, pp. 600-609, 2015.

[11] M. H. Soreshjani, “Direct torque controlled space vector modulated method for line start permanent magnet synchronous and induction motors,” *Trans. Inst. Meas. Control*, vol. 37, no. 6, pp. 826-840, 2015.
[12] L. Yousfi, A. Bouchemha, M. Bechouat, and A. Boukrouche, “Vector control of induction machine using PI controller optimized by genetic algorithms,” Proc. of 16th International Power Electronics and Motion Control Conference and Exposition, pp. 1272-1277, 2014.

[13] Y. Li, K. H. Ang, and G. C. Y. Chong, “PID control system analysis and design,” IEEE Control Syst. Mag., vol. 26, no. 1, pp. 32-41, 2006.

[14] S. Hajji, A. Ayadi, Y. A. Zorgani, T. Maatoug, M. Farza, and M. M’Saad, “Integral backstepping-based output feedback controller for the induction motor,” Trans. Inst. Meas. Control, vol. 41, no. 16, pp. 4599-4612, 2019.

[15] M. W. Dunnigan, S. Wade, B. W. Williams, and X. Yu, “Position control of a vector controlled induction machine using Slotine’s sliding mode control approach,” IEEE Proc. Electr. Power Appl., vol. 145, no. 3, pp. 231-238, 1998.

[16] L. Wang, S. Chai, D. Yoo, L. Gan, and K. Ng, PID and Predictive Control of Electrical Drives and Power Converters using Matlab®/Simulink®, 2014.

[17] Y. Zhang and H. Yang, “Model predictive torque control of induction motor drives with optimal duty cycle control,” IEEE Trans. Power Electron., vol. 29, no. 12, pp. 6593-6603, 2014.

[18] R. Uddin and J. Ryu, “Predictive control approaches for bilateral teleoperation.” Annu. Rev. Control, vol. 42, pp. 82-99, 2016.

[19] E. Daryabeigi and C. Lucas, “Simultaneously, speed and flux control of an induction motor, with brain emotional learning based intelligent controller (BELBIC),” Proc. of IEEE International Electric Machines and Drives Conference, pp. 1085-1092, 2009.

[20] M. Fatourechi, C. Lucas, and A. Khaki Sedigh, Emotional Learning as a New Tool for Development of Agent-based Systems, vol. 27, no. 2. 2003.

[21] C. Lucas, D. Shahmirzadi, and N. Sheikholeslami, “Introducing belbic: Brain emotional learning based intelligent controller,” Intell. Autom. Soft Comput., vol. 10, no. 1, pp. 11-21, 2004.

[22] H. Rouhani, M. Jalili, B. N. Araabi, W. Eppler, and C. Lucas, “Brain emotional learning based intelligent controller applied to neurofuzzy model of micro-heat exchanger,” Expert Syst. Appl., vol. 32, no. 3, pp. 911-918, 2007.

[23] M. Fatourechi, C. Lucas, and A. K. Sedigh, “Reducing control effort by means of emotional learning,” Proceedings of 9th Iranian Conference on Electrical Engineering (ICEE2001), p. 41, 2001.

[24] M. A. Hannan, J. A. Ali, P. J. Ker, A. Mohamed, M. S. H. Lipu, and A. Hussain, “Switching techniques and intelligent controllers for induction motor drive: Issues and recommendations,” IEEE Access, vol. 6, pp. 47489-47510, 2018.

[25] J. Wang, F. Wang, G. Wang, S. Li, and L. Yu, “Generalized proportional integral observer based robust finite control set predictive current control for induction motor systems with time-varying disturbances,” IEEE Trans. Ind. Informatics, vol. 14, no. 9, pp. 4159-4168, 2018.
Muhammad Affan received his B.E. degree in electrical engineering from the Department of Electrical Engineering at NED University of Engineering and Technology, Karachi, Pakistan. His current research interests include model-based and model-free control design for electrical drives and robotic applications.

Riaz Uddin received his B.E. and M.E. degrees in electrical engineering from the Department of Electrical Engineering at NED University of Engineering and Technology, Karachi, Pakistan, in 2005 and 2008, respectively. He received his Ph.D. degree from the School of Mechatronics, Gwangju Institute of Science and Technology (GIST), Gwangju, Korea, in 2016. He joined NED as a lecturer in 2005 and now he is working as an Assistant Professor in the Department of Electrical Engineering & Director of the Office of Research, Innovation and Commercialization in NED University of Engineering and Technology. He is also the PI/Director of Haptics, Human-Robotics and Condition Monitoring Lab affiliated Lab of National Center of Robotics and Automation (NCRA), HEC/PC, Pakistan. He recently developed an indigenous ICU-ventilator during COVID-19 pandemic. His research interests include control systems, automation, instrumentation, smart systems, computer networked Systems, robotics, haptics and teleoperation.

Publisher's Note Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.