Turkish Journal of Electrical Engineering & Computer Sciences

http://journals.tubitak.gov.tr/elektrik/

Research Article

Turk J Elec Eng & Comp Sci
(2017) 25: 2968 – 2985
© TÜBİTAK
doi:10.3906/elk-1608-115

# MOPSO-based predictive control strategy for efficient operation of sensorless vector-controlled fuel cell electric vehicle induction motor drives

**Adel Abdelaziz Abdelghany ELGAMMAL**[1,*], **Mohamed Fathy EL-NAGGAR**[2,3]
[1]Department of Energy, University of Trinidad and Tobago, Point Lisas Campus, Couva, Trinidad and Tobago
[2]Department of Electrical Engineering, Prince Sattam bin Abdulaziz University, Al Kharj, Saudi Arabia
[3]Department of Electrical Engineering, Helwan University, Helwan, Egypt

**Received:** 11.08.2016 • **Accepted/Published Online:** 14.11.2016 • **Final Version:** 30.07.2017

**Abstract:** This paper introduces an optimal control strategy of model-based predictive control (MPC) based on multiobjective particle swarm optimization (MOPSO) for a sensorless vector control induction motor, which is used in a fuel cell electric vehicle drive system. The proposed MPC-MOPSO algorithm is implemented to tune the weighting parameters of the MPC controller to tackle all the conflicting objective functions. The paper handles the following fitness functions: minimizing the speed error, minimizing the torque ripple, minimizing the DC-link voltage ripple, and minimizing machine flux ripple. Computer simulations studies have been completed utilizing MATLAB/Simulink with a specific end goal of assessing the dynamic performance of the proposed MPC-MOPSO optimal controller and comparing it with single-objective particle swarm optimization and traditional PI controllers. The simulation results demonstrate the good dynamic response of the proposed MPC-MOPSO optimal tuning strategy over the traditional PI controllers for more accurate tracking performance through the whole speed range, especially at starting conditions and load change disturbances.

**Key words:** Electric vehicle, fuel cell, sensorless vector control, multiobjective particle swarm optimization, model-based predictive control

## 1. Introduction

Concern about air contamination, global warming, and the quick consumption of fossil fuels such as oil, gas, and coal has prompted the proposition of fuel cell electric vehicles (FC-EVs) to replace traditional vehicles [1]. Usually, batteries are utilized for all types of EVs as a second rechargeable energy source to implement good acceleration and regenerative braking while the main energy source such as the FC is utilized to give extended driving range [2]. In spite of the fact that FCs show great power ability in steady-state operation, the technical restrictions of FCs such as low productivity in a low load request, moderate power transfer rate in transient operation, and high cost for each watt make the reaction of the FC in transient and momentary crest power requests generally poor. In this way, batteries and/or superconductors are utilized with the FC. However, the batteries introduce a few downsides, such as short lifecycle, long charging time, and low power densities, despite giving noteworthy high-energy density potential, and they might be enormously influenced by high current discharges [3]. The induction motors that are used in the EVs could be controlled by field-oriented control (FOC) to operate at their optimum efficiency, torque, and speed. In the last few years, FOC drives have

[*]Correspondence: adel.elgammal@utt.edu.tt

been widely utilized for induction motor optimal speed control [4]. Because of their simple control structure and inexpensive cost, PID and hysteresis current controllers are commonly used in traditional FOC to achieve smooth transient operation of the induction motor drive [5]. In any case, the PI controller demonstrates poor transient reaction against system parametric variety and sudden load change influence, particularly during low-speed operations. The trend in industrial applications is to implement sensorless vector control techniques to estimate the motor speed and flux using computational methods instead of using speed sensors to strengthen the drive system reliability [6]. The generally utilized strategies for speed observation are the model reference adaptive system [7,8], neural networks [9], extended Kalman filter [10,11], and nonlinear observer [12]. A comprehensive literature review of optimal control of induction motor drives based on sensorless vector control was given in [13]; however, new solutions should be suggested to overcome a few issues connected with sensorless vector control [14]. Current research works focus on wide speed ranges, especially very low speed ranges [15], or inaccuracy and IM parameter variations [16]. Intelligent techniques such as neural networks, fuzzy control, and genetic algorithms have been utilized to handle the nonlinearities and uncertainties of the IM parameters [17–20]. Model-based predictive control (MPC) has become a major control technology due to its ability to deal with a dynamic optimization problem with physical constraints that come from the industrial applications and control process [21]. MPC relies on the dynamic models of the process to compute a vector of optimal control signals by solving an optimization problem online considering the process constraints to minimize the future signal error, which is the difference between the desired signal and the future outputs predicted by the process model. At that point, the first signal of the optimal input vector is connected to the plant and this strategy from the prediction and optimization with new optimal input is repeated in the following sampling interval [22]. Comparing the performance of MPC to conventional PID controllers, the main disadvantage of MPC is that it is a generally more perplexing controller and it requires more time for online prediction, particularly when constraints exist. However, the literature has shown that the performance usually indicates obvious MPC predominance [23]. MPC has been utilized in optimal speed/torque control for induction motor drive systems [24] and joined with speed/flux estimators [25]. Recently, FLC control and PSO approaches have been implemented in the automatic tuning of model predictive control parameters [26–28]. In this paper, the MPC-multiobjective particle swarm optimization (MOPSO) optimal control approach is implemented to control the speed, torque, flux, and DC voltage for a sensorless vector control IM drive, which is intended to be implemented in a FC-EV drive system. The proposed MPC-MOPSO algorithm is implemented to tune the weighting parameters of the MPC controller to handle the following fitness functions: minimizing the speed error, minimizing the torque ripple, minimizing the DC-link voltage ripple, and minimizing machine flux ripple while considering the provided drive system constraints. The control signal is predicted by MPC based on the current control variables, the actual signal error, and the predicted future output of the model. The dynamic performance of the MPC-MOPSO controller is tested with different speed tracks and load disturbance and compared using MATLAB/Simulink with the traditional PI controller and single-objective particle swarm optimization (SOPSO). SOPSO and MOPSO can be utilized to figure out this multiobjective optimization problem. In SOPSO, a single weighted objective function may be formed from all fitness functions to obtain a single solution, while MOPSO finds a set of acceptable trade-off optimal solutions, which is called the Pareto front. The main point of interest in MOPSO is that it does not need a priori knowledge about the relative significance of the targets [29–32]. This paper is organized as follows: Section 2 introduces the dynamic model of the induction motor. Section 3 outlines the fundamental structure of the model-based predictive control. Section 4 presents the system arrangement of a typical fuel cell/supercapacitor/battery EV. Section 5 demonstrates the proposed MOPSO and SOPSO optimal algorithms. Section 6 shows the dynamic simulations using MATLAB/Simulink. Finally, particular important conclusions are presented in Section 7.

## 2. The sensorless control of the IM

The dynamic state-space model of the IM can be represented using the $d-q$ reference frame as follows [33]:

$$\frac{d}{dt}\begin{bmatrix} i_{ds} \\ i_{qs} \\ \lambda_{dr} \\ \lambda_{qr} \end{bmatrix} = \begin{bmatrix} -\gamma & \omega_s & \eta\mu & \omega_r\mu \\ -\omega_s & -\gamma & -\omega_r\mu & \eta\mu \\ \eta L_m & 0 & -\eta & \omega_{sl} \\ 0 & \eta L_m & -\omega_{sl} & -\eta \end{bmatrix}\begin{bmatrix} i_{ds} \\ i_{qs} \\ \lambda_{dr} \\ \lambda_{qr} \end{bmatrix} + \begin{bmatrix} \frac{1}{\sigma L_s} & 0 \\ 0 & \frac{1}{\sigma L_s} \\ 0 & 0 \\ 0 & 0 \end{bmatrix}\begin{bmatrix} v_{ds} \\ v_{qs} \\ 0 \\ 0 \end{bmatrix}, \tag{1}$$

$$T_e = \frac{3PL_m}{4L_r}\left(i_{qs}\lambda_{dr} - i_{ds}\lambda_{qr}\right), \tag{2}$$

where:

$$\begin{aligned} &\sigma = 1 - \tfrac{L_m^2}{L_sL_r};\ \gamma = \tfrac{1}{\sigma L_s}\left(R_s + \tfrac{L_m^2}{L_r^2}R_r\right)\ ; \\ &\eta = \tfrac{R_r}{L_r};\ \mu = \tfrac{L_m}{\sigma L_sL_r};\ \omega_{sl} = \omega_s - \omega_r \end{aligned} \tag{3}$$

$v_{ds}$ and $v_{qs}$ are $d-q$stator voltage components, $i_{ds}$ and $i_{qs}$ are $d-q$stator current components, $\lambda_{dr}$ and $\lambda_{qr}$ are $d-q$rotor flux linkage components, $\omega_s$ is stator angular speed, $\omega_r$ is mechanical rotor speed, $\omega_{sl}$ is slip angular speed, $T_e$ is the electromagnetic torque, $P$ is the number of pole pairs, $R_s$ and $R_r$ are stator and rotor resistance referred to the stator, and $L_s$, $L_r$, and $L_m$ are stator, rotor, and mutual inductances referred to the stator. The speed can be evaluated by the model referencing adaptive system, which primarily comprises the reference and adaptive models. The adjustment strategy creates the assessed speed where the calculated values of the reference model are compared with the output of the adjustable model until the difference between the two models reaches zero. The reference model represents the stator voltage equation in the stator reference frame, which can be composed as:

$$\frac{d}{dt}\begin{bmatrix} \lambda_{dr}^s \\ \lambda_{qr}^s \end{bmatrix} = \frac{L_r}{L_m}\begin{bmatrix} v_{ds}^s \\ v_{qs}^s \end{bmatrix} - \frac{L_r}{L_m}\begin{bmatrix} \left(R_s + \sigma L_s\frac{d}{dt}\right) & 0 \\ 0 & \left(R_s + \sigma L_s\frac{d}{dt}\right) \end{bmatrix}\begin{bmatrix} i_{ds}^s \\ i_{qs}^s \end{bmatrix}. \tag{4}$$

The adaptive model represents the rotor voltage equation of the IM in the stator reference frame, which can be written as:

$$\frac{d}{dt}\begin{bmatrix} \hat{\lambda}_{dr}^s \\ \hat{\lambda}_{qr}^s \end{bmatrix} = \begin{bmatrix} -\frac{1}{T_r} & -\hat{\omega}_r \\ \hat{\omega}_r & -\frac{1}{T_r} \end{bmatrix}\begin{bmatrix} \lambda_{dr}^s \\ \lambda_{qr}^s \end{bmatrix} + \frac{L_m}{T_r}\begin{bmatrix} i_{ds}^s \\ i_{qs}^s \end{bmatrix}, \tag{5}$$

where $T_r$ is the rotor circuit time constant, $\hat{\omega}_r$ is the adaptive rotor speed, and $\hat{\lambda}_{dr}^s$ and $\hat{\lambda}_{qr}^s$ are the estimated $d-q$ rotor flux linkages in the stationary reference frame.

The signal of the speed tuning error $\varepsilon$can be calculated as:

$$\varepsilon = \hat{\lambda}_{dr}^s\lambda_{qr}^s - \hat{\lambda}_{qr}^s\lambda_{dr}^s. \tag{6}$$

With the correct estimated rotor speed, the two flux linkages calculated by the two models should coordinate. An adjustment calculation with the PI control can be utilized to tune the speed until the two flux values coordinate:

$$\hat{\omega}_r = \left(K_p + \frac{K_i}{S}\right)\left(\hat{\lambda}^s_{dr}\lambda^s_{qr} - \hat{\lambda}^s_{qr}\lambda^s_{dr}\right). \tag{7}$$

## 3. Model-based predictive control

MPC is a type of control that utilizes the accurate model of the plant to predict the future performance of a process. The control action is obtained by solving a constrained optimization problem at every sample interval to minimize the difference between the estimated future outputs and the reference value through the use of minimum control energy and satisfying the constraints of the plant, as shown in Figure 1. There are mainly four available components:

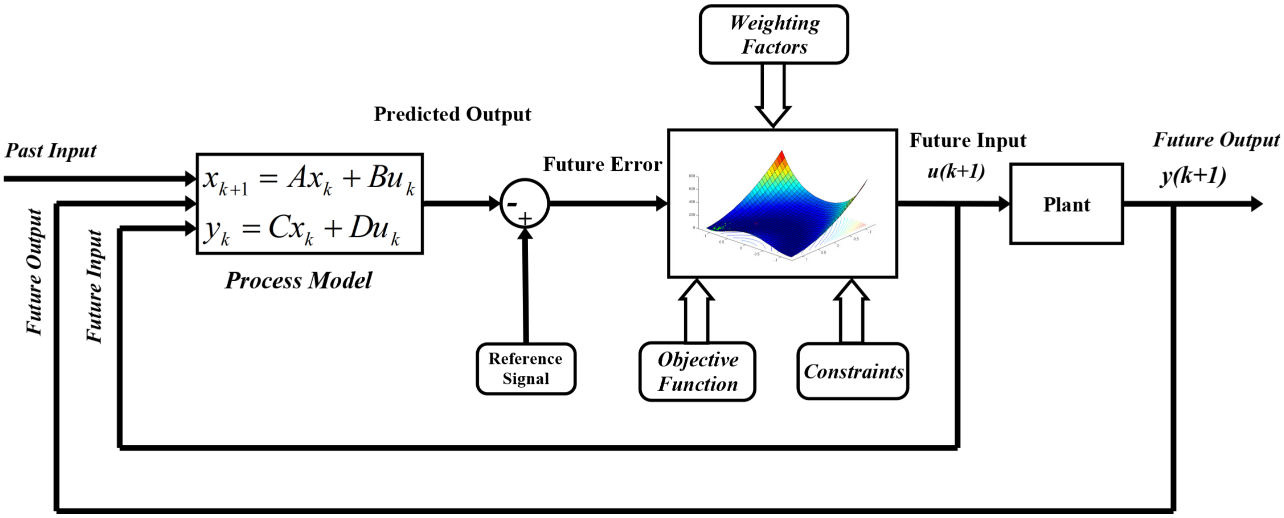

**Figure 1.** General structure of model predictive controller.

1. The process model: for each instance k, the system model is utilized to predict the future response of the controlled plant (y (k), y (k + 1), y (k + 2), . . . , y (k + N$_P$ − 1)) over a specified prediction horizon N$_P$. In MPC it is assumed that the model can be formed as a state-space model, as shown in the following equation:

$$\begin{aligned} x_{k+1} &= Ax_k + Bu_k \\ y_k &= Cx_k + Du_k \end{aligned}. \tag{8}$$

1. The cost function: for each iteration k, the MPC controller calculates a sequence of future control actions (u (k), u (k + 1), u (k + 2),. . . , u (k + N$_c$ − 1)) over a specified control horizon N$_c$ such that the objective function is optimized (minimized or maximized) subject to the specified constraints. The objective function is generally a combination of weighted sum of square predicted errors and square future control values.

2. The optimizer to optimize the future plant behavior by computing at each time instance k the optimal value u (k) of the control vector, which is implemented to the system: at the next sample time k + 1, the whole process of prediction and optimization will be repeated.

3. Process constraints: constraints can be either on the control variable or on the input control signals. The first-order Euler approximations can be utilized to derive the discrete model of the IM as follows:

$$T_s \begin{bmatrix} i_{ds} \\ i_{qs} \\ \lambda_{dr} \\ \lambda_{qr} \end{bmatrix} = \begin{bmatrix} i_{ds}(k+1) \\ i_{qs}(k+1) \\ \lambda_{dr}(k+1) \\ \lambda_{qr}(k+1) \end{bmatrix} - \begin{bmatrix} i_{ds}(k) \\ i_{qs}(k) \\ \lambda_{dr}(k) \\ \lambda_{qr}(k) \end{bmatrix}, \tag{9}$$

where T$_s$ is the sampling time. The control variables for the (k + 1)th sampling moment can be predicted as follows:

$$\begin{bmatrix} i_{ds}(k+1) \\ i_{qs}(k+1) \\ \lambda_{dr}(k+1) \\ \lambda_{qr}(k+1) \end{bmatrix} = \begin{bmatrix} i_{ds}(k) \\ [1ex] i_{qs}(k) \\ \lambda_{dr}(k) \\ \lambda_{qr}(k) \end{bmatrix} + T_s \begin{bmatrix} -\gamma & \omega_s & \eta\mu & \omega_r\mu \\ -\omega_s & -\gamma & -\omega_r\mu & \eta\mu \\ \eta L_m & 0 & -\eta & \omega_{sl} \\ 0 & \eta L_m & -\omega_{sl} & -\eta \end{bmatrix} \begin{bmatrix} i_{ds}(k) \\ i_{qs}(k) \\ \lambda_{dr}(k) \\ \lambda_{qr}(k) \end{bmatrix} + \begin{bmatrix} \frac{1}{\sigma L_s} & 0 \\ 0 & \frac{1}{\sigma L_s} \\ 0 & 0 \\ 0 & 0 \end{bmatrix} \begin{bmatrix} v_{ds}(k+1) \\ v_{qs}(k+1) \\ 0 \\ 0 \end{bmatrix}, \tag{10}$$

$$T_e(k+1) = \frac{3PL_m}{4L_r}\left(i_{qs}(k+1)\lambda_{dr}(k+1) - i_{ds}(k+1)\lambda_{qr}(k+1)\right). \tag{11}$$

Similarly, for the second sampling moment, (k + 2), the control variables can be predicted using the following equations:

$$\begin{bmatrix} i_{ds}(k+2) \\ i_{qs}(k+2) \\ \lambda_{dr}(k+2) \\ \lambda_{qr}(k+2) \end{bmatrix} = \begin{bmatrix} i_{ds}(k+1) \\ i_{qs}(k+1) \\ \lambda_{dr}(k+1) \\ \lambda_{qr}(k+1) \end{bmatrix} + T_s \begin{bmatrix} -\gamma & \omega_s & \eta\mu & \omega_r\mu \\ -\omega_s & -\gamma & -\omega_r\mu & \eta\mu \\ \eta L_m & 0 & -\eta & \omega_{sl} \\ 0 & \eta L_m & -\omega_{sl} & -\eta \end{bmatrix} \begin{bmatrix} i_{ds}(k+1) \\ i_{qs}(k+1) \\ \lambda_{dr}(k+1) \\ \lambda_{qr}(k+1) \end{bmatrix} + \begin{bmatrix} \frac{1}{\sigma L_s} & 0 \\ 0 & \frac{1}{\sigma L_s} \\ 0 & 0 \\ 0 & 0 \end{bmatrix} \begin{bmatrix} v_{ds}(k+2) \\ v_{qs}(k+2) \\ 0 \\ 0 \end{bmatrix}, \tag{12}$$

$$T_e(k+2) = \frac{3PL_m}{4L_r}\left(i_{qs}(k+2)\lambda_{dr}(k+2) - i_{ds}(k+2)\lambda_{qr}(k+2)\right). \tag{13}$$

## 4. System configuration

A combination of different energy sources, a primary energy source such as the FC stack and one or more secondary energy sources such as batteries and supercapacitors, is used to supply the EV drive system as shown in Figure 2. The essential reason for using batteries and supercapacitors with the FC stack is to satisfy the different energy requirements, especially the start-up and transient energy requirements. Figure 3 presents the configuration of the proposed sensorless vector control based on the MPC-MOPSO controller including a voltage source inverter, hysteresis current controller, coordinate transformers, and flux and speed estimator. In this drive system, only the three-phase motor current should be measured and the motor speed and flux are calculated by a closed-loop estimator. The three-phase currents are measured and expressed by two coordinates, $i\alpha$ and$i\beta$, using the Clarke transformation module. The Park transformation is applied to transform $(\alpha, \beta)$ in the rotating reference frame $(d, q)$. Four estimated and measured variables ($\omega$, T, $\lambda$, V$_{DC}$) are compared with the reference values and fed to the MPC-MOPSO controller in order to produce the $d-q$ current commands$i^*_{ds}$ and $i^*_{qs}$.

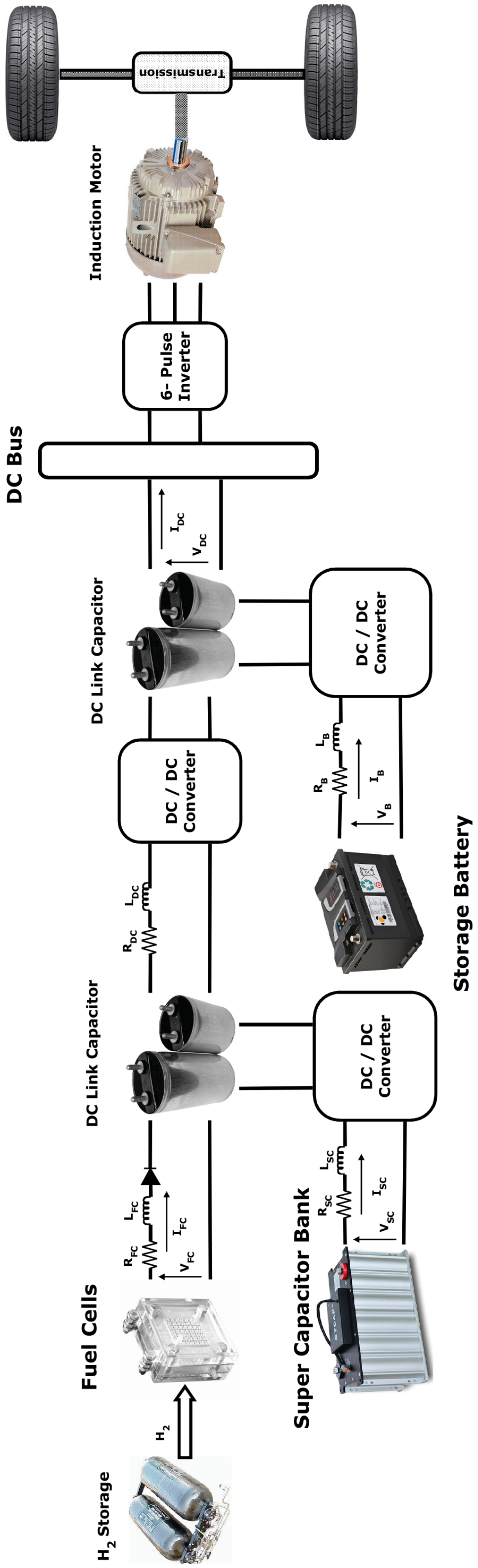

**Figure 2.** The configuration of the proposed FC/supercapacitor/battery EV.

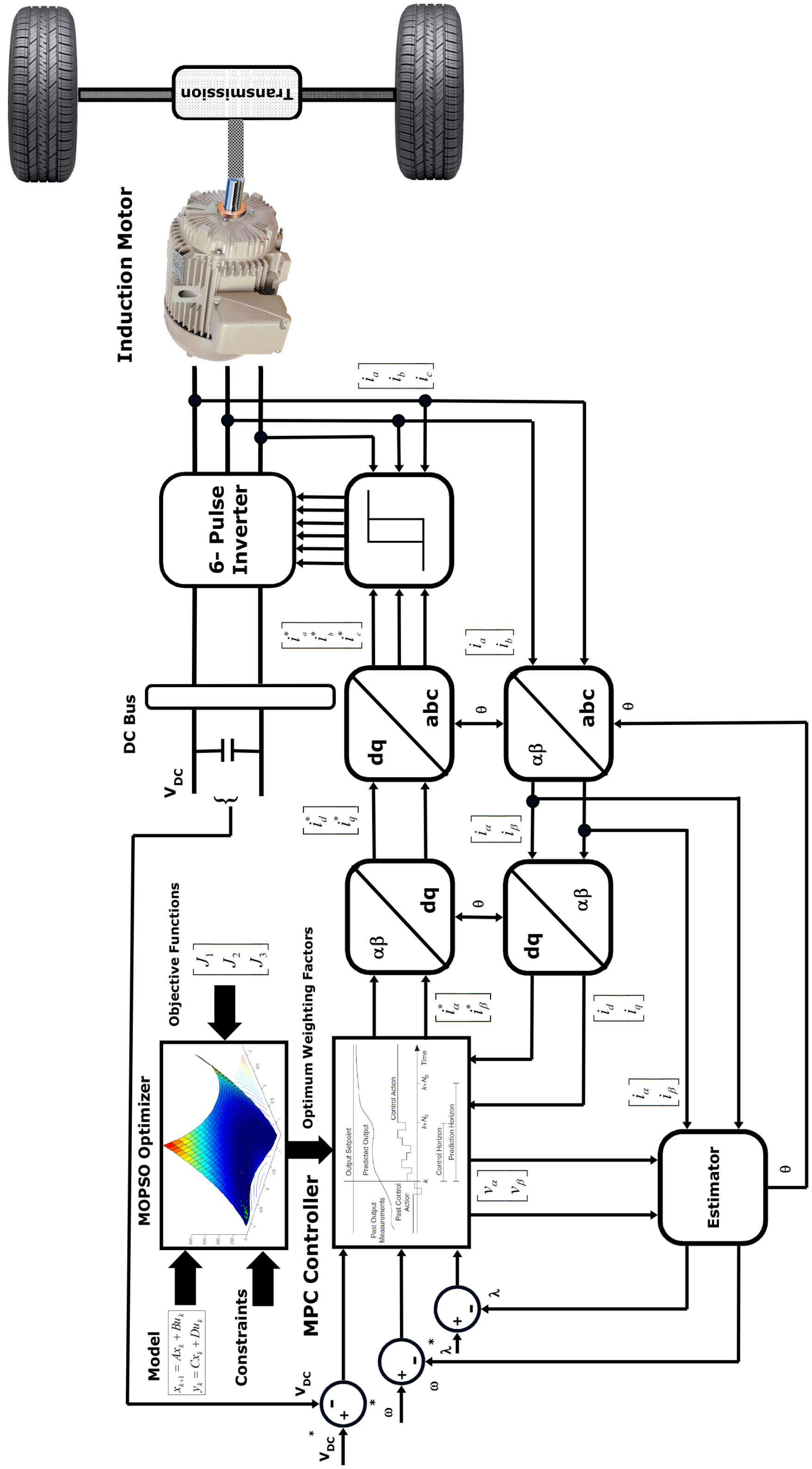

**Figure 3.** Structure of the proposed sensorless vector control of the IM using MPC-MOPSO.

## 5. Proposed MPC-MOPSO controller

The MPC-MOPSO control strategy has been suggested to regulate the stator flux, the speed, the torque of the IM, and the DC-link voltage. All these control objectives can be combined in a single weighted objective function as follows:

1) To minimize the speed error:

$$J_1(k) = \sum_{k=0}^{N_P} (\omega_{ref}(k) - \omega(k))^2. \tag{14}$$

2) To minimize the torque ripple:

$$J_2(k) = \sum_{k=0}^{N_P} (T_{ref}(k) - T(k))^2. \tag{15}$$

3) To minimize the harmonic content of the DC voltage:

$$J_3(k) = \sum_{k=0}^{N_P} (V_{DC-ref}(k) - V_{DC}(k))^2. \tag{16}$$

4) To minimize the error on the stator flux:

$$J_4(k) = \sum_{k=0}^{N_P} (\lambda_{ref}(k) - \lambda(k))^2. \tag{17}$$

Subject to:

$$|T_e(t)| \leq T_{\max}, \tag{18}$$

$$\sqrt{i_{ds}^2 + i_{qs}^2} \leq I_{s,\max}, \tag{19}$$

$$\sqrt{v_{ds}^2 + v_{qs}^2} \leq V_{s,\max}, \tag{20}$$

$$0 \leq V_{DC} \leq V_{DC,\max}, \tag{21}$$

where I$_{s;\max}$ and V$_{s;\max}$ are the limits of the stator current and voltage, V$_{dc;\max}$ is the physical limit of the DC-link voltage, and T$_{\max}$ is the limit of the electromagnetic torque. To obtain minimum solutions for this complex optimality search problem, a single combined weighted objective function could be applied to aggregate the features of all the stated objective functions as follows:

$$J_o = k_1 J_1 + k_2 J_2 + k_3 J_3 + k_4 J_4, \tag{22}$$

where k$_1$, k$_2$, k$_3$, and k$_4$ are the weighting parameters. Therefore, the decision variables can be defined as a vector, i.e. $\mathbf{x} = \{K_1, K_2, K_3, K_4\}$ . SOPSO is utilized to find the optimal weighting coefficients $\mathbf{x} = \{K_1, K_2, K_3, K_4\}$ to optimize the single aggregate objective function. Improper selection of the weighting coefficients will result in solutions that would not satisfy all the objectives. Therefore, MOPSO is utilized to produce a set of trade-off optimal solutions (Pareto front), which give more ability to the user to see a wide range of optimal solutions.

### 5.1. The SOPSO algorithm

The main procedure of the SOPSO algorithm and the determination process is characterized as follows:

1. Assign the maximum and minimum values of the four weighting variables ($k_1, k_2, k_3, k_4$), the quantity of particles (swarm size = 20 particles), and the maximum number of iterations as end basis (100 iterations).

$$\begin{aligned} k_{1\,\min} \le k_1 \le k_{1\,\max} \\ k_{2\,\min} \le k_2 \le k_{2\,\max} \\ k_{3\,\min} \le k_3 \le k_{3\,\max} \\ k_{4\,\min} \le k_4 \le k_{4\,\max} \end{aligned} \tag{23}$$

2. The particles' solutions, velocities, and best positions $P_{best}$ are initialized by assigning a population of random values and all particles start to fly in all dimensions.

3. These initial values for the weighting parameters are sent to the MPC to assess the dynamic performance of the drive system. For each solution, $P(k_1, k_2, k_3, k_4)$, the single objective weighted function of Eq. (22) is calculated.

4. Each particle's solution is compared with the previous best solution, *pbest*. If the particle's solution is better than the previous best solution, *pbest* is assigned as the particle's solution. All particles' best solutions are compared with each other and the global best solution *gbest* is assigned as the best solution.

5. Update the velocity v of each particle for each dimension using the following equation:

$$\begin{pmatrix} v_{i1}(t+1) \\ v_{i2}(t+1) \\ v_{i3}(t+1) \\ v_{i4}(t+1) \end{pmatrix} = \omega \times \begin{pmatrix} v_{i1}(t) \\ v_{i2}(t) \\ v_{i3}(t) \\ v_{i4}(t) \end{pmatrix} + c_1 \times r_1() \times \begin{pmatrix} pbest_{i1}(t) - k_{i1}(t) \\ pbest_{i2}(t) - k_{i2}(t) \\ pbest_{i3}(t) - k_{i3}(t) \\ pbest_{i4}(t) - k_{i4}(t) \end{pmatrix} + c_2 \times r_2() \times \begin{pmatrix} gbest_{i1}(t) - k_{i1}(t) \\ gbest_{i2}(t) - k_{i2}(t) \\ gbest_{i3}(t) - k_{i3}(t) \\ gbest_{i4}(t) - k_{i4}(t) \end{pmatrix}. \\ where: \quad i = 1, 2, ..., n = 20 \tag{24}$$

6. The maximum and minimum limits of the velocity of each particle for each dimension are checked using the following equation:

$$\begin{array}{l} if v_{id}(t+1) \ge v_d^{\max} then v_{id}(t+1) = v_d^{\max} \\ if v_{id}(t+1) \le v_d^{\min} then v_{id}(t+1) = v_d^{\min} \end{array}. \tag{25}$$

7. Update the solution of each particle using the following equation:

$$\begin{pmatrix} k_{i1}(t+1) \\ k_{i2}(t+1) \\ k_{i3}(t+1) \\ k_{i4}(t+1) \end{pmatrix} = \begin{pmatrix} k_{i1}(t) \\ k_{i2}(t) \\ k_{i3}(t) \\ k_{i4}(t) \end{pmatrix} + \begin{pmatrix} v_{i1}(t+1) \\ v_{i2}(t+1) \\ v_{i3}(t+1) \\ v_{i4}(t+1) \end{pmatrix}, \tag{26}$$

$$\mathrm{X_{id}{=}X_{id}{+}V_{id}} \quad \begin{array}{l} where: k_1^{\min} \le k_{1i}(t+1) \le k_1^{\max} \\ k_2^{\min} \le k_{2i}(t+1) \le k_2^{\max} \\ k_3^{\min} \le k_{3i}(t+1) \le k_3^{\max} \\ k_4^{\min} \le k_{4i}(t+1) \le k_4^{\max} \end{array}. \tag{27}$$

8. Steps 3 to 7 are repeated until the convergence criterion is met and the optimal weighting parameters are generated by the latest *gbest*.

### 5.2. The MOPSO algorithm

The main strategy of the MOPSO algorithm and the selection practice is characterized as follows:

1. Initialize all particles' positions and velocities with random solutions satisfying all constraints and limits shown in Eqs. (23) and (25).

2. These population solutions are sent to the MPC to simulate the drive system behavior utilizing the objective functions as shown in Eqs. (14)–(17) and are saved in *pbest* vector form.

3. Calculate the multiobjective fitness values of each particle and store all nondominated solutions in the Pareto archive.

4. The velocity of each particle is updated using the following equation:

$$\begin{pmatrix} v_{i1}(t+1) \\ v_{i2}(t+1) \\ v_{i3}(t+1) \\ v_{i4}(t+1) \end{pmatrix} = \omega \times \begin{pmatrix} v_{i1}(t) \\ v_{i2}(t) \\ v_{i3}(t) \\ v_{i4}(t) \end{pmatrix} + c_1 \times r_1() \times \begin{pmatrix} p_{p1}(t) - k_{i1}(t) \\ p_{p2}(t) - k_{i2}(t) \\ p_{p3}(t) - k_{i3}(t) \\ p_{p4}(t) - k_{i4}(t) \end{pmatrix} + c_2 \times r_2() \times \begin{pmatrix} p_{r1}(t) - k_{i1}(t) \\ p_{r2}(t) - k_{i2}(t) \\ p_{r3}(t) - k_{i3}(t) \\ p_{r4}(t) - k_{i4}(t) \end{pmatrix}, \quad (28)$$
$$where: \quad i = 1, 2, ..., n = 20$$

where $\mathrm{P_{id}}, \mathrm{P_{gd}} P_{r,d}\, and\ P_{p,d}$ are randomly chosen from the Pareto archive.

5. The solution of each particle is updated using Eq. (26). The particles' values are kept in the allowable range shown in Eq. (27).

6. Evaluate the multiobjective function value of each particle and store the nondominated particles in the Pareto archive.

7. Two particles' solutions are selected randomly from the Pareto archive for $\mathrm{P}_{r,d}$, $\mathrm{P}_{i,d}$.

8. Repeat Steps 3 to 7 until the convergence conditions are reached.

## 6. Simulation results

Computer simulations have been completed utilizing MATLAB/Simulink programming to validate the performance of the IM, which is optimally controlled using the proposed MPC-MOPSO control approach. The nominal parameters of the induction motor are as follows: rated power = 3.75 kW, rated voltage = 380 V, rated frequency = 60 Hz, stator resistance $\mathrm{R}_s = 1.235\ \Omega$, stator inductance $\mathrm{L}_s = 0.1175$ H, rotor resistance $\mathrm{R}_r = 0.857\ \Omega$, rotor inductance $\mathrm{L}_r = 0.0964$ H, mutual inductance $\mathrm{L}_m = 0.0635$ H, pole pairs = 4, inertia = 0.058 kgm$^2$. The suitable selection of prediction horizon (Np) and control horizon (Nc) will enhance the system dynamic performance and increase the accuracy of the predicted values. In the proposed system, these values are selected as $\mathrm{N}_p = 70$ and $\mathrm{N}_c = 5$. The dynamic system performance of the proposed MPC-MOPSO control strategy is compared with MPC-SOPSO and the conventional PI controller to show the effectiveness of MOPSO-MPC. The parameters of the PI controller have been selected as $\mathrm{K}_p = 9$ and $\mathrm{K}_i = 25$. The parameters of the PI controller were tuned using Skogestad's model-based method with smaller integral time for faster

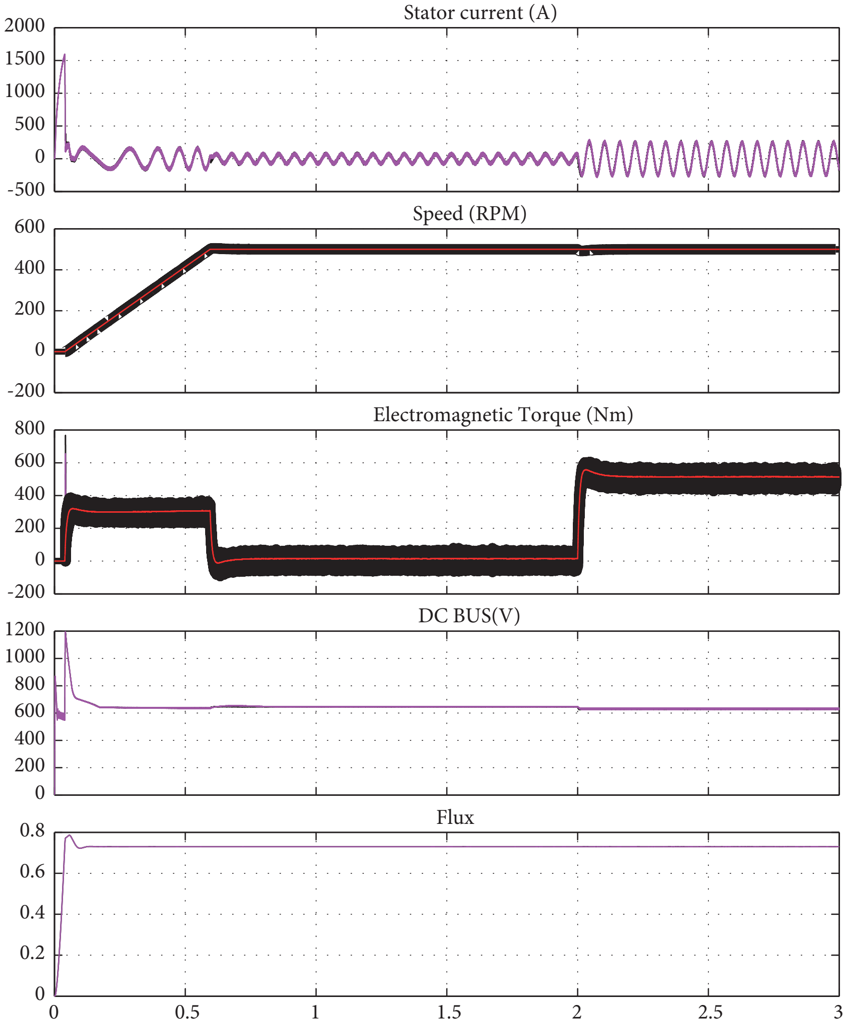

**Figure 4.** Transient response of the IM drive using MOPSO-MPC controller for the first speed track assuming load torque disturbance. The load torque is stepped up from 0.0 Nm to 50 Nm at $t = 2$ s.

disturbance compensation [34]. In the first speed trajectory, the speed of the IM is kept at zero for 0.1 s and then the speed increased to 500 rpm in 0.5 s, and then the motor speed is kept constant at this value during the next 2.5 s. The load torque is stepped up from 0.0 Nm to 50 Nm at $t = 2$ s. It has been observed that the speed command and actual motor speed are adjusted and good tracking performance has been accomplished utilizing MOPSO-MPC. A small speed plunge is seen at the moment of load disturbance; however, the motor speed is reestablished very rapidly. Figures 4–6 illustrate the dynamic performance of the MPC-MOPSO control strategy and the MPC-SOPSO versus the conventional PI controllers. Clearly, the PI controller has poor dynamic performance, particularly during the starting condition, and it is significantly influenced when a load disturbance is implemented. The MPC-MOPSO has been tried likewise at different speed commands, including negative speed, very low speed, and zero speed. In the second speed trajectory, the machine starts from zero speed increased straight to 500 rpm at time t = 0.6 s, and then the speed is kept constant until t = 1 s. At that point the speed of the IM is decelerated back to zero at t =1.6 s and then the speed is kept steady at zero until $t = 2$ s, and then the speed is accelerated in the reverse direction straight to –500 rpm at $t = 2.6$ s. The speed is

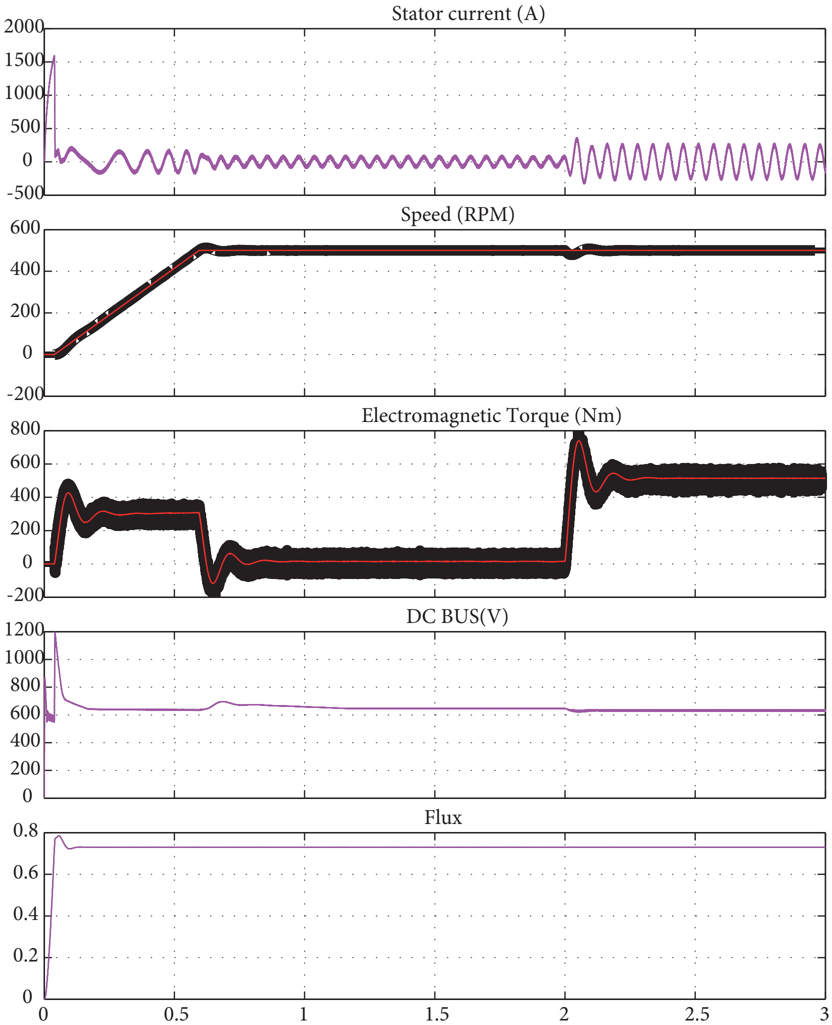

**Figure 5.** Transient response of the IM drive using PID controller for the first speed track assuming load torque disturbance. The load torque is stepped up from 0.0 Nm to 50 Nm at $t = 2$ s.

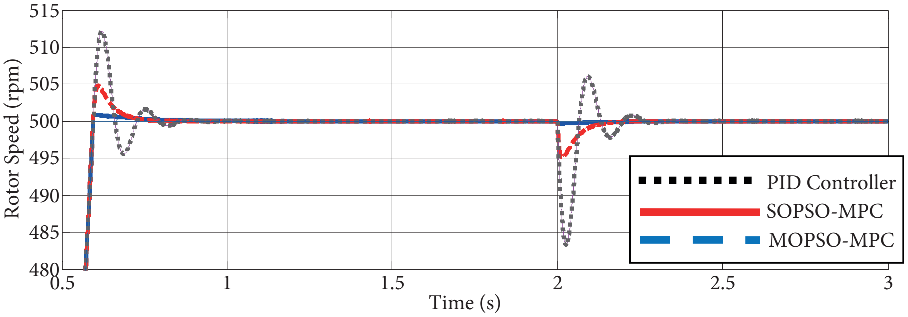

**Figure 6.** Comparison of speed dynamic responses for the first speed track with step load disturbance (the load torque is stepped up from 0.0 Nm to 50 Nm at $t = 2$ s) based on the proposed MOPSO-MPC, SOPSO-MPC, and PI controllers. Zooming for different instances at 0.5 and 2.0 s.

then kept constant at –500 rpm until $t = 3$ s. The load torque is suddenly applied from zero to 50 Nm at $t = 2$ s. Figures 7–9 demonstrate the comparison of the dynamic performance utilizing the MPC-MOPSO, MPC-

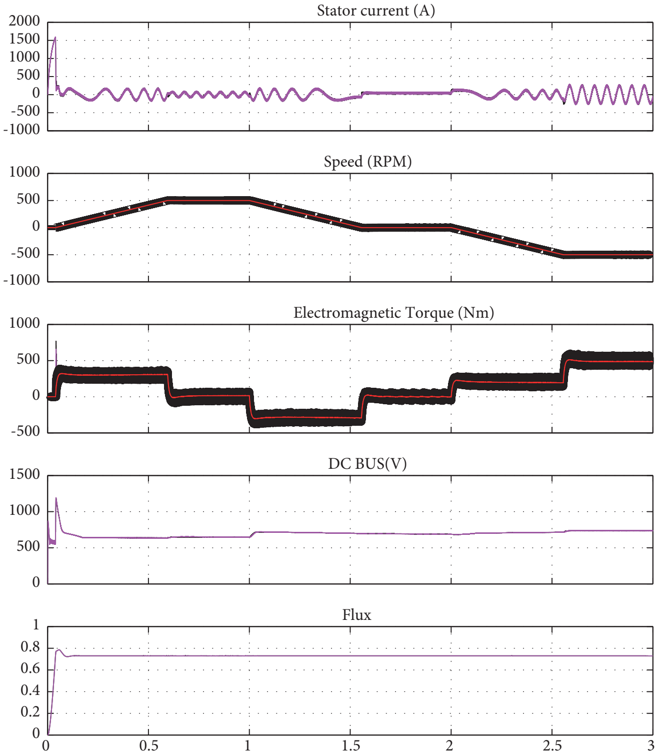

**Figure 7.** Transient response of the IM drive using MOPSO-MPC controller for the second speed track assuming load torque disturbance. The load torque is stepped up from 0.0 Nm to 50 Nm at $t = 2$ s.

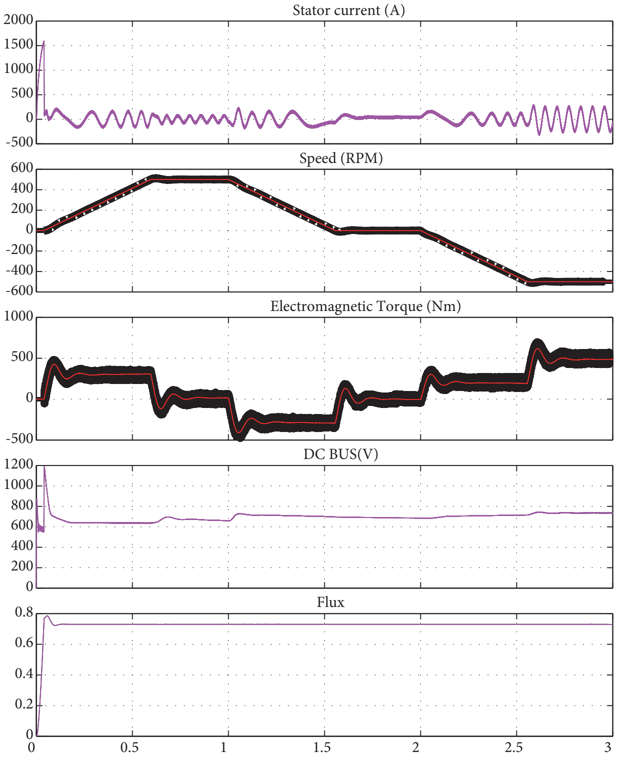

**Figure 8.** Transient response of the IM drive using PI controller for the second speed track assuming load torque disturbance. The load torque is stepped up from 0.0 Nm to 50 Nm at $t = 2$ s.

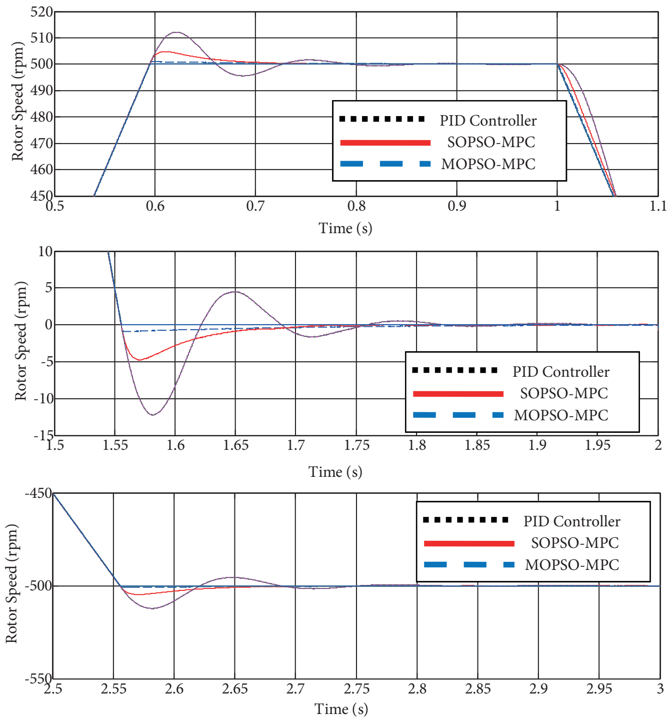

**Figure 9.** Comparison of speed dynamic responses for the second speed track with step load disturbance (the load torque is stepped up from 0.0 Nm to 50 Nm at $t = 2$ s) based on the proposed MOPSO-MPC, SOPSO-MPC, and PI controllers. Zooming for different instances at 0.6, 1, 1.6, and 2.6 s.

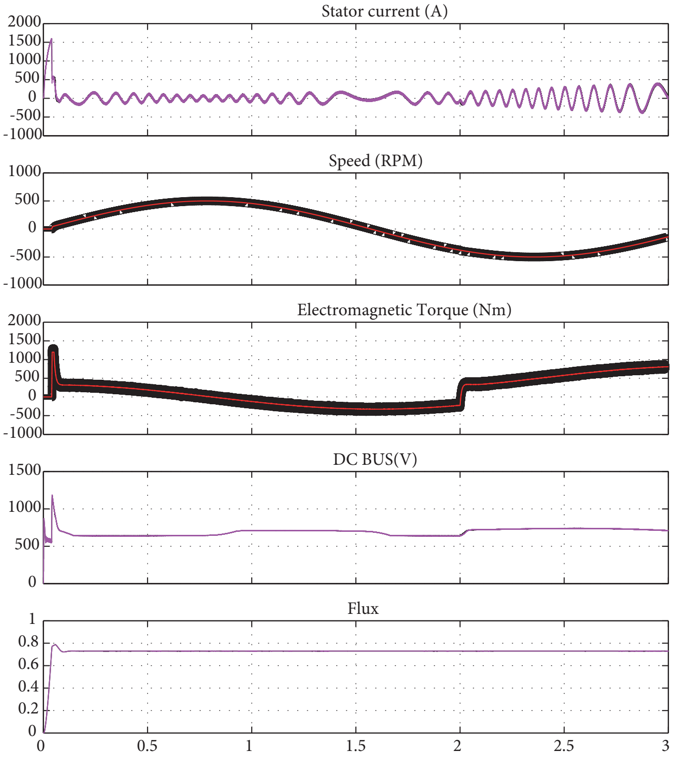

**Figure 10.** Transient response of the IM drive using MOPSO-MPC controller for the third speed track assuming load torque disturbance. The load torque is stepped up from 0.0 Nm to 50 Nm at $t = 2$ s.

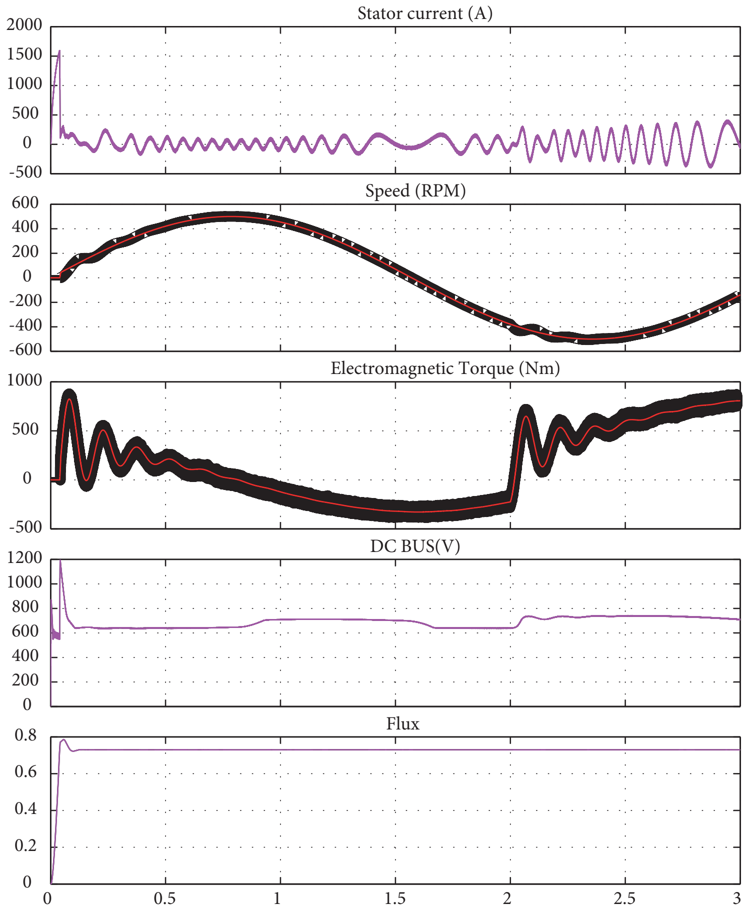

**Figure 11.** Transient response of the IM drive using PI controller for the third speed track assuming load torque disturbance. The load torque is stepped up from 0.0 Nm to 50 Nm at $t = 2$ s.

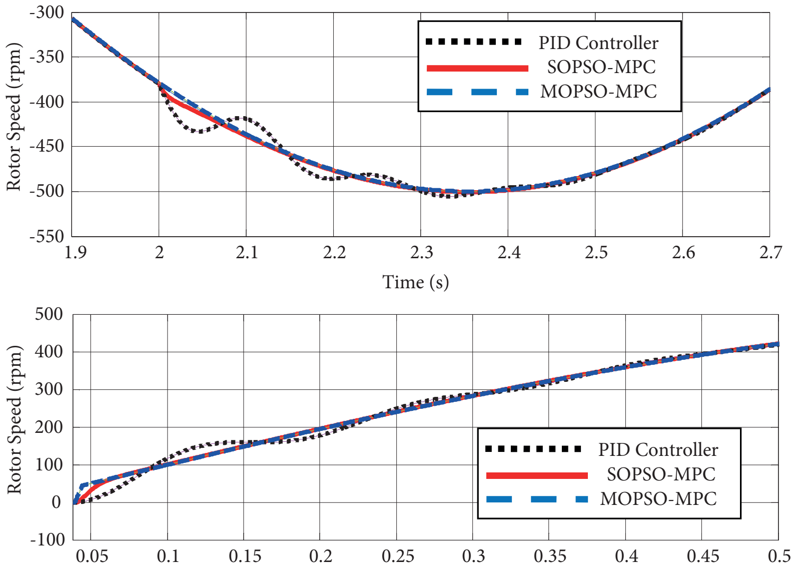

**Figure 12.** Comparison of speed dynamic responses for the third speed track with step load disturbance (the load torque is stepped up from 0.0 Nm to 50 Nm at $t = 2$ s) based on the proposed MOPSO-MPC, SOPSO-MPC, and PI controllers. Zooming for starting and step load instances at $t = 2$ s.

SOPSO, and PI controllers. It has been demonstrated that the proposed MPC-MOPSO controller has better dynamic performance and has accomplished the steady-state value very rapidly. Although there is some speed variety, the speed comes back to its typical level rapidly when the step-load torque is applied, showing solid effectiveness against load torque disturbance while the PI controller showed a bad performance particularly at starting conditions and at the moment of load disturbance or speed change. Additionally, the dynamic response of the proposed MPC-MOPSO controller has been tested using MPC-MOPSO and PI controllers utilizing a sinusoidal speed trajectory with 500 rpm amplitude and the load torque is suddenly increased from no load to 50 Nm at $t = 2$ s. Figures 10–12 demonstrate that an excellent tracking response has been accomplished using the MPC-MOPSO control strategy in spite of the load torque disturbance. The results illustrate the capacity of MPC-MOPSO to accurately follow the sinusoidal speed track and to handle the load torque disturbance. Interestingly, unmistakably poor transient performance is connected with the PI controller in the beginning period and when load aggravation is connected at $t = 2$ s. A THD and error performance comparison is given in the Table for the proposed MOPSO-MPC controller, SOPSO-MPC controller, and conventional PI controller. From the results given in the Table, it is evident that the proposed MOPSO-MPC can improve the performance of the IM and lessen the effects of load and speed disturbances. It is clear that the THD of the stator current is reduced from 7.76% (using the conventional PID controller) and 3.65% (using the SOPSO-MPC controller) to 1.79% (using the MOPSO-MPC controller), which is below the mark of 5% specified in the IEEE-519 standard. The proposed MOPSO-MPC controller has yielded the lowest THD (1.79%) and has better performance as compared with the other controllers.

**Table.** System performance comparison.

| Control technique | Conventional PID controller | SOPSO-MPC | MOPSO-MPC |
|---|---|---|---|
| Speed error (%) | 7.76% | 3.65% | 1.79% |
| THDi (stator current) | 6.18% | 2.93% | 1.24% |
| Torque ripple (%) | 7.26% | 3.75% | 2.59% |
| THDv (inverter voltage) | 5.16% | 3.02% | 1.16% |
| THD DC bus voltage | 8.12% | 4.08% | 2.14% |
| DC bus voltage error % | 5.76% | 4.05% | 2.09% |

## 7. Conclusion

This paper has effectively exhibited the utilization of a new control approach based on MPC-MOPSO for optimal control of constrained sensorless vector control of an induction motor, which is utilized as a part of a FC-battery-superconductor EV. A set of multiobjective functions has been handled using the proposed MPC-MOPSO while respecting the provided drive system constraints appointed to the load torque, the magnetic flux, the DC voltage, and the motor speed. The effectiveness of the proposed MPC-MOPSO optimization control approach has been assessed at different operating speeds and load torque conditions such as forward, reverse, very low, high speed, and sinusoidal speed track and a load torque disturbance was applied for each speed command with machine parameter variations. The simulation results carried out by MATLAB/Simulink demonstrated that the proposed MPC-MOPSO controller has a better dynamic response and better tracking of speed trajectory at all speeds than both the SOPSO and PI controllers. The proposed MPC algorithm can achieve satisfactory performance when appropriate parameters are chosen and tuned using MOPSO and hence this can be applied to large fields of industrial process controllers.

## References

[1] Abdallah T, Mamadou BC, Brayima D, Yacine A. DC/DC and DC/AC converters control for hybrid electric vehicles energy management: ultracapacitors and fuel cell. IEEE T Ind Inform 2013; 9: 686-696.

[2] Thounthong P, Pierfederici S, Martin JP, Hinaje M, Davat B. Modeling and control of fuel cell/supercapacitor hybrid source based on differential flatness control. IEEE T Veh Technol 2010; 59: 2700-2710.

[3] Camara MB, Dakyo B, Gualous H. Supercapacitors and battery energy management based on new European driving cycle. Journal of Energy and Power Engineering 2012; 6: 168-177.

[4] Ahmed ZD, Vladimir P. Vector controlled induction motor drive based on model predictive control. In: 11th International Conference on Actual Problems of Electronics Instrument Engineering; 2–4 October 2012; Novosibirsk, Russia. pp. 167-173.

[5] Jose T, Jayendra T, Kamalesh H, Krishna V. An improved scheme for extended power loss ride-through in a voltage-source-inverter-fed vector-controlled induction motor drive using a loss minimization technique. IEEE T Ind Appl 2016; 52: 1500-1508.

[6] Jaroslaw G, Haitham A. Speed sensorless induction motor drive with predictive current controller. IEEE T Ind Electron 2013; 60: 699-709.

[7] Andrew S, Shady G, John F. Improved rotor flux estimation at low speeds for torque MRAS-based sensorless induction motor drives. IEEE T Energy Conver 2016; 31: 270-282.

[8] Yaman Z, Shady G, David A. Model predictive MRAS estimator for sensorless induction motor drives. IEEE T Ind Electron 2016; 63: 3511-3521.

[9] Xiaodong S, Long C, Zebin Y, Huangqiu Z. speed-sensorless vector control of a bearingless induction motor with artificial neural network inverse speed observer. IEEE-ASME T Mech 2013; 18: 1357-1366.

[10] Francesco A, Tommaso C, Filippo D, Adriano F, Antonino S. Convergence analysis of extended Kalman filter for sensorless control on induction motor. IEEE T Ind Electron 2015; 62: 2341-2352.

[11] Habibullah MD, Dylan DL. A speed-sensorless FS-PTC of induction motors using extended Kalman filters. IEEE T Ind Electron 2015; 62: 6765-6778.

[12] Mousavi-Aghdam SR, Sharifian MBB. Nonlinear adaptive observer for sensorless control of induction motor. In: 20th Iranian Conference on Electrical Engineering; 2012: Tehran, Iran. pp. 376-379.

[13] Yung-Chang L, Chen-Lung T, Wen-Cheng P, Cheng-Tao T. Full-order stator flux observer based sensorless vector controlled induction motor drives applying particle swarm optimization algorithm. In: International Symposium on Computer, Consumer and Control; 4–6 July 2016; Xi'an, China. pp. 899-902.

[14] Patel C, Ramchand R, Sivakumar K, Das A, Gopakumar K. A rotor flux estimation during zero and active vector periods using current error space vector from a hysteresis controller for a sensorless vector control of IM drive. IEEE T Ind Electron 2011; 58: 2334-2344.

[15] Lascu C, Boldea I, Blaabjerg F. A class of speed-sensorless sliding mode observers for high-performance induction motor drives. IEEE T Ind Electron 2009; 56: 3394-3403.

[16] Zaky MS. Stability analysis of speed and stator resistance estimators for sensorless induction motor drives. IEEE T Ind Electron 2012; 59: 858-870.

[17] Boglietti A, Cavagnino A, Lazzari M. Computational algorithms for induction-motor equivalent circuit parameter determination—Part I: Resistances and leakage reactances. IEEE T Ind Electron 2011; 58: 3723-3733.

[18] Ko J, Choi J, Chung D. Hybrid artificial intelligent control for speed control of induction motor. In: International Joint Conference SICE-ICASE; 18–21 October 2006; Busan, South Korea. pp. 678-683.

[19] Ji ZC, Shen YX. Back-stepping position control for induction motor based on neural network. In: 1st IEEE Conference on Industrial Electronics and Applications; 24–26 May 2006; Singapore. New York, NY, USA: IEEE. pp. 1-5.

[20] Lin FJ, Shieh HJ, Shyu KK, Huang PK. On-line gain tuning IP controller using real coded genetic algorithm. Electr Pow Syst Res 2004; 72: 157-169.

[21] Khalil M, Ahmed E, Mohamed R. Model predictive control using FPGA. International Journal of Control Theory and Computer Modeling 2015; 5: 1-14.

[22] Sivakumar R, Shennes M. Design and development of model predictive controller for binary distillation column. International Journal of Science and Research 2014; 5: 445-451.

[23] Holkar K, Waghmare L. An overview of model predictive control. International Journal of Control and Automation 2010; 3: 47-63.

[24] Thomsen S, Hoffmann N, Fuchs F. PI control, PI-Based state space control and model-based predictive control for drive systems with elastically coupled loads-a comparative study. IEEE T Ind Electron 2011; 58: 3647-3657.

[25] Mapok K, Zuva T, Masebu H, Zuva K. Performance comparison of two controllers on a nonlinear system. International Journal of Chaos, Control, Modelling and Simulation 2013; 2: 17-30.

[26] Bayoumi EHE, Awadallah MA, Soliman HM. Deadbeat performance of vector-controlled induction motor drives using particle swarm optimization and adaptive neuro-fuzzy inference systems. Electro-motion Scientific Journal 2011; 18: 231-242.

[27] Ryohei S, Fukiko K, Hideyuki I, Chikashi N, Yoshikazu F, Eitaro A. Automatic tuning of model predictive control using particle swarm optimization. In: Proceedings of the IEEE Swarm Intelligence Symposium; 2007. New York, NY, USA: IEEE. pp. 221-226.

[28] Sofiane B, Mohammed C, Fouad A, Salim F. A new approach for fuzzy predictive adaptive controller design using particle swarm optimization algorithm. Int J Innov Comput I 2013; 9: 3741-3758.

[29] Ngatchou P, Zarei A, El-Sharkawi A. Pareto multi objective optimization. In: Proceedings of the 13th International Conference on Intelligent Systems Application to Power Systems; 6–10 November 2005. pp. 84 -91.

[30] Berizzi A, Innorta M, Marannino P. Multiobjective optimization techniques applied to modern power systems. In: IEEE Power Engineering Society Winter Meeting; 2001. New York, NY, USA: IEEE. pp. 1503-1508.

[31] Coello CA, Lechuga MS. MOPSO: A proposal for multiple objective particle swarm optimization. In: Proceedings of the 2002. Congress on Evolutionary Computation, Part of the 2002 IEEE World Congress on Computational Intelligence; May 2002. New York, NY, USA: IEEE. pp. 1051-1056.

[32] Jihane K, Mohamed C. Optimization of hybrid renewable energy power systems using evolutionary algorithms. In: 5th International Conference on Systems and Control; 25–27 May 2016; Marrakesh, Morocco. pp. 383-388.

[33] Fawzan S, Mohamed A. Model predictive control for deadbeat performance of induction motor drives. WSEAS Transactions on Circuits and Systems 2015; 14: 304-312.

[34] Finn H. Comparing PI tuning methods in a real benchmark temperature control system. Norwegian Society of Automatic Control, Modeling, Identification and Control 2010; 31: 79-91.