Multi-Objective Particle Swarm Optimization Approach for Field-Oriented Control of Linear Induction Motor Considering End-Effect Phenomena

Khettache Laid[1*], Djarah Djalal [2], Rezoug Mohamed Redha[3], Benmakhlouf Abdeslam[4]

[1,2,3,4]Department of Electrical Engineering, Faculty of Applied Sciences Kasdi Merbah Ouargla University, Algeria

E-mail : khettache.laid@univ-ouargla.dz (corresponding)

djarah.djalal@univ-ouargla.dz, mr.rezoug@univ-ouargla.dz, benmakhlouf.abdeslam@univ-ouargla.dz

Abstract: In modern electric drive systems, precise control of the motor is crucial for achieving high performance and energy efficiency. Field-Oriented Control (FOC) is widely adopted due to its ability to decouple torque and flux, but traditional tuning methods for cascaded speed, current, and flux control loops remain a challenging task due to multiple conflicts among these loops. Furthermore, accurately controlling Linear Induction Motors (LIMs) is limited and complicated by the presence of longitudinal end effects due to their open magnetic circuit. To overcome these limitations, this paper proposes a novel approach employing Multi-Objective Particle Swarm Optimization (MOPSO) to automatically and systematically optimize the parameters of the FOC strategy. In the proposed approach (MOPSO), each PI controller corresponding to the speed, current, and flux control loops is initially tuned using Particle Swarm Optimization (PSO) with ITAE as the objective; Then, these locally optimized parameters are collectively refined using the multi-objective optimization process (MO) to ensure the minimization of transient performance indicators such as overshoot, settling time, rise time, and steady-state error. The effectiveness of the MOPSO-tuned FOC is evaluated through detailed MATLAB/Simulink simulations under various dynamic operating conditions, including step changes in speed reference and sudden load variations. Moreover, comparative simulation analyses demonstrate that the MOPSO-tuned FOC significantly reduces overshoot by 3.98%, shortens rise by 56.34%, and settling times by 38.13%, and confirms its superiority in maintaining control precision, including excellent reference tracking and disturbance rejection.

Keywords: Linear induction motor, PI tuning, moultiobjective, Particle Swarm Optimization, field-oriented control, end-effect

1 Introduction

In recent years, the demand for improved efficiency, precision, and direct motion control in many industries has increased the importance of Linear Induction Motors (LIMs) [1]. Unlike traditional motors that require gears or mechanical components to produce linear motion, LIMs deliver linear motion directly through electromagnetic thrust. This inherent design makes them faster, more reliable, and easier to maintain. However, achieving precise and robust control of LIMs presents unique challenges due to their open magnetic circuit, longitudinal end-effects, and inherent nonlinearities [2]. The rotor of a linear induction motor (LIM) is typically a solid conducting plate or strip that is mounted or placed close to the stator; this open magnetic circuit causes magnetic field distortion at the ends of the motor's primary, called the end-effect phenomenon [3]. Thus, these characteristics pose a significant challenge for effective control strategies. Various design techniques, optimization strategies, and control algorithms can be used to mitigate the effects of the end effect and enhance the overall performance of LIMs in practical applications [4] [5] [6] [7]. Among these methods, Field-Oriented Control (FOC) is a high-performance control strategy widely used in electric motors and has emerged as a powerful technique for achieving decoupled control of thrust and flux in LIMs, analogous to the independent control in separately excited DC motors [8]. The FOC framework typically involves cascaded control loops: inner current control loops (d-q axis), an outer speed control loop, and a flux control loop, often employing Proportional-Integral (PI) controllers [9]. The performance of the FOC scheme is critically dependent on the appropriate tuning of the parameters within these loops. Conventional methods for tuning Field-Oriented Control (FOC) controller parameters frequently depend on classical control theory, such as the pole placement method, or manual iterative adjustments. Furthermore, these classical methods can be time-consuming, may not attain globally optimal performance under varying operating conditions, and may encounter difficulties in simultaneously satisfying multiple performance objectives, particularly in the presence of complex Linear Induction Motor (LIM) dynamics. To overcome these limitations, intelligent optimization techniques, particularly those capable of handling multi-objective problems, offer a promising avenue for automated and systematic controller parameter tuning [10] [11] [12]. In contrast, conventional single-objective optimization techniques, when applied to such systems, frequently yield sub-optimal solutions due to the inherent trade-offs among conflicting performance metrics. For instance, an enhancement in speed response might inadvertently compromise current ripple or flux stability. To systematically address this multi-objective optimization challenge, a robust evolutionary computation technique was required. Hence, the Multi-Objective Particle Swarm Optimization (MOPSO) algorithm was selected for its efficacy in solving complex multi-objective problems across various engineering fields [13]. MOPSO leverages a population-based search mechanism combined with the concept of Pareto dominance to simultaneously optimize multiple conflicting performance criteria.

This approach yields a set of non-dominated solutions that represent the best trade-offs among competing objectives. The primary contribution of this study lies in the introduction of a dual-stage optimization approach specifically engineered for LIM control under end-effect conditions. In the first stage, each PI controller corresponding to the speed, current, and flux control loops is initially tuned using Particle Swarm Optimization (PSO), with the tuning process guided by the Integral of Time-weighted Absolute Error (ITAE) criterion. which ensures the minimization of transient performance indicators such as overshoot, settling time, rise time, and steady-state error. In the second stage, these locally optimized parameters are then collectively refined using the multi-objective optimization process (MO), to resolve the inherent conflicts between speed tracking, flux stability, and current regulation. This coordinated optimization explicitly resolves inter-loop interactions and performance trade-offs, resulting in a Pareto front of non-dominated solutions that enables systematic trade-off analysis between transient responsiveness and steady-state accuracy, a capability not achievable with conventional empirical or single-objective tuning approaches. The remainder of the paper is structured as follows: Section 2 presents the mathematical modelling of the Linear Induction Motor considering the end-effect and the associated field-oriented control scheme. Section 3 details the approach used for the formulation of the optimization problem, the multi-objective tuning of the PI controllers, and the defined performance objectives. Section 4 discusses the simulation results in MATLAB/Simulink software, performs a comparative analysis with conventional methods, and evaluates the derived Pareto front. Finally, Section 5 provides the concluding remarks and outlines future research directions.

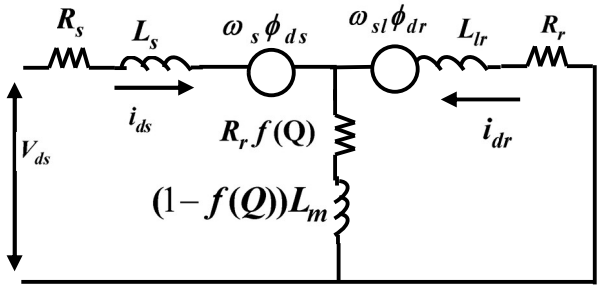
2 *Control Strategy*

To effectively analyze and control the complex dynamic behavior of Linear Induction Motors, particularly in the presence of end effects, a robust mathematical framework is indispensable. The d-q synchronous reference frame modelling approach has proven highly effective in simplifying the analysis of AC machines by transforming time-varying parameters into DC quantities under steady-state conditions. Besides that, the modelling of the LIM-based FOC control strategy relies on two fundamental steps:

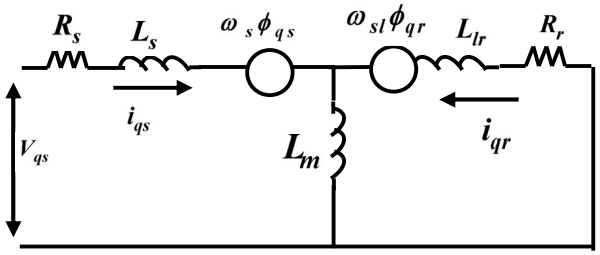
2.1. Model of LIM with end-effect

This step consists in establishing the dynamic equations of the motor through the d-q equivalent electrical circuit, which explicitly incorporates the phenomenon of the end effect. The equivalent circuit of a LIM (see Figure1), when considering end effects, differs from that of a conventional rotary induction motor. As depicted in the relevant literature [14, 15], this circuit can generally be divided into two distinct

parts. The first is the d-axis equivalent circuit, which is significantly influenced by the end effect; this phenomenon causes parameters such as resistance and inductance to vary and distorts the magnetic field at the motor's entry and exit points. In contrast, the q-axis equivalent circuit remains largely unaffected by the end effect and closely resembles that of a standard induction motor.



a-Direct - axis equivalent circuit



b- Quadrature- axis equivalent circuit

Figure 1

LIM equivalent electrical circuit

In [16] [17], a critical parameter for quantifying the end-effect phenomenon in LIMs is denoted as Q and defined by:

$$Q = \frac{D.R_r}{(L_{lr} + L_m).v} \tag{1}$$

This equation highlights Q dependence on the rotor resistance (Rr), magnetizing inductance (Lm), secondary leakage inductance (L_{lr}), and crucially, the linear velocity (v) of the motor.

Q is inversely dependent on velocity; at zero velocity, the primary's effective length can be considered infinite, and end effects become negligible. However, as the velocity increases, Q also increases, signifying a more pronounced end effect. This

increase in end effects directly causes a reduction in the LIM's magnetization current [18].

The influence of Q is further represented by a function f(Q), which modifies key motor parameters and is defined as follows:

$$f(Q)=\frac{1-e^{-Q}}{Q} \tag{2}$$

Thus, the three-phase magnetizing inductance ($\hat{L}_m$) varies with Q, effectively accounting for the demagnetizing influence of the end effect, as given by the following expression:

$$\hat{L}_m = L_m\left[1-f(Q)\right] \tag{3}$$

Furthermore, when eddy current losses are considered, a resistance emerges in the transverse branch of the equivalent circuit, denoted as $\hat{R}_r$. This resistance is also dependent on *f(Q)* can be calculated as follows:

$$\hat{R}_r = R_r f(Q) \tag{4}$$

Based on the previous equations and the motor's equivalent circuit, the voltage and flux linkage equations can be expressed as:

$$\begin{cases} V_{ds} = R_s i_{ds} + R_r f(Q)\left(i_{ds}+i_{dr}\right)+\dfrac{d\phi_{ds}}{dt}-\omega_s\phi_{qs} \\ V_{qs} = R_s i_{qs} + \dfrac{d\phi_{qs}}{dt}+\omega_s\phi_{ds} \\ 0 = R_r i_{dr} + R_r f(Q)\left(i_{ds}+i_{dr}\right)+\dfrac{d\phi_{dr}}{dt}-\omega_{sl}\phi_{qr} \\ 0 = R_r i_{qr} + \dfrac{d\phi_{qr}}{dt}+\omega_{sl}\phi_{dr} \end{cases} \tag{5}$$

$$\begin{cases} \phi_{ds} = L_{ls} i_{ds} + L_m\left(1-f(Q)\right)\left(i_{ds}+i_{dr}\right) \\ \phi_{qs} = L_{ls} i_{qs} + L_m\left(i_{ds}+i_{dr}\right) \\ \phi_{dr} = L_{lr} i_{dr} + L_m\left(1-f(Q)\right)\left(i_{ds}+i_{dr}\right) \\ \phi_{qr} = L_{lr} i_{qr} + L_m\left(i_{ds}+i_{dr}\right) \end{cases} \tag{6}$$

In these flux linkage equations, the term (*1−f(Q)*) directly modifies the magnetizing inductance component in the d-axis flux linkages (ϕds and ϕdr), providing a mathematical representation of the end-effect's impact on the magnetic coupling.

The primary angular frequency (*ωs*) is determined by combining the linor (rotor) angular speed (*ωr*) with the slip frequency (*ωsl*):

$$\omega_s = \omega_{sl} + \omega_r \tag{7}$$

The conversion of the LIM's angular speed to linear speed (v) is given by:

$$v = \frac{\tau_p}{\pi}\omega \tag{8}$$

The electromagnetic force (Fe) can be defined in terms of d-q axis fluxes and currents:

$$F_e = \frac{3\pi}{2\tau_p}\frac{p}{2}\left(\phi_{ds} i_{qs} - \phi_{qs} i_{ds}\right) \tag{9}$$

An alternative expression for the electromagnetic force, explicitly incorporating the end-effect function $f(Q)$, is also provided:

$$F_e = \frac{3\pi}{2\tau_p}\frac{p}{2}\frac{L_m\left(1-f\left(Q\right)\right)}{L_r - L_m f\left(Q\right)}\left(\phi_{dr} i_{qs} - \phi_{qr} i_{ds}\right) \tag{10}$$

The dynamic equation governing the linear motion of the LIM is expressed as:

$$F_e = Mv + B\dot{v} + F_L \tag{11}$$

Here, M represents the mass of the moving parts, B is the damping coefficient and F_L is the load force.

2.2. Field-Oriented Control (FOC)

Once the motor model has been established, the FOC control is designed to decouple the control of flux and thrust, similar to how torque and flux are controlled in rotary machines. Furthermore, the principle of decoupled control will be elaborated, highlighting how the d-axis current component primarily controls the magnetic flux, while the q-axis current component primarily controls the thrust (force) developed by the LIM. In analogy to a separately excited DC motor, the FOC scheme (Figure. 2) typically comprises inner current control loops for regulating id and iq to their reference values (id* and iq*), as well as an outer speed control loop that generates the reference q-axis current (iq*) based on the speed error, and a flux control loop that generates the reference d-axis current (id*) to maintain the desired magnetic flux [19].

For effective decoupling in FOC, specific conditions for rotor flux orientation are established within the d-q synchronous reference frame. These conditions significantly simplify the control structure:

$$\begin{cases} \phi_{qr} = 0 \\ \phi_{dr} = \phi_r = constant \end{cases} \tag{12}$$

By setting the q-axis rotor flux (ϕqr) to zero, the control system ensures that the d-axis rotor flux (ϕdr) remains constant and equal to the total rotor flux (ϕr). Under these conditions, the linor (rotor) current components can be derived as follows [20]:

$$\begin{cases} i_{dr} = \dfrac{\phi_r - \hat{L}_m . i_{ds}}{L_r} \\ i_{qr} = \dfrac{-L_m . i_{qs}}{L_{lr} + L_m} \end{cases} \tag{13}$$

By integrated the rotor flux orientation conditions (Equation 12) and the derived linor currents (Equations 13) into the stator voltage equations (Equation 5), the decoupling of current and voltage components is expressed as:

$$\begin{cases} V_{ds} = \left(R_s + \hat{R}_r - \dfrac{\hat{R}_r . \hat{L}_m}{L_r} \right) i_{ds} + \dfrac{\hat{R}_r}{L_r} \phi_r + \sigma L_s \dfrac{di_{ds}}{dt} - \omega_s \sigma' i_{ds} \\ V_{qs} = R_s i_{qs} + \sigma' \dfrac{di_{qs}}{dt} + \omega_s \sigma L_s i_{ds} + \omega_s \dfrac{\hat{L}_m}{L_r} \phi_r \end{cases} \tag{14}$$

These equations reveal the inherent coupling between the d- and q-axis voltage components, primarily due to back electromotive force (EMF) terms:

$$\begin{cases} E_d = -\dfrac{\hat{R}_r}{L_r} \phi_r + \omega_s \sigma' i_{ds} \\ E_q = -\omega_s \sigma L_s i_{ds} - \omega_s \dfrac{\hat{L}_m}{L_r} \phi_r \end{cases} \tag{15}$$

Moreover, the thrust force can be represented in a simplified form, making it more amenable to control:

$$F_e^* = K_f . i_{qs}^* \tag{16}$$

$$K_f = \frac{3\pi}{2\tau_p} \frac{p}{2} \frac{L_m \left(1 - f\left(Q\right)\right)}{L_r - L_m f\left(Q\right)} \phi_{dr}^* \tag{17}$$

This equation clearly demonstrates how the end effect influences the thrust constant. From this, the desired torque-producing current (iqs*) can be derived based on the desired thrust:

$$i_{qs}^* = \frac{4\tau_p}{3\pi p} \frac{L_r - L_m f\left(Q\right)}{L_m \left(1 - f\left(Q\right)\right)} \frac{F_e^*}{\phi_{dr}^*} \tag{18}$$

2.3. Pole Placement Tuning of PI Controllers

In this study, the initial determination of the PI controller parameters is performed using the Pole Placement Method. This classical control approach is used to establish a theoretically stable baseline for the speed, flux, and current regulators by mapping the closed-loop poles of the system to desired locations on the complex s-plane. By defining specific performance criteria, such as the damping ratio (ζ) and natural frequency (ω_n), this approach provides a mathematically rigorous starting point that ensures the system operates within a stable region, [20] [21]. Each control loop is modeled according to its specific physical time constants, and the gains are calculated by mapping the closed-loop poles to a desired characteristic equation:

$$s^2 + 2\zeta\omega_n s + \omega_n{}^2 = 0 \tag{19}$$

2.3.1. Inner Current Loops (d-q axis)

The electrical dynamics of the LIM are represented by the stator resistance Rs and the transient inductance L_σ. The baseline gains are computed as follows:

$$\begin{cases} K_{pd,q} = 2\zeta\omega_{nc}L_\sigma - R_s \\ K_{id,q} = L_\sigma\omega_{nc}{}^2 \end{cases} \tag{20}$$

2.3.2. Flux Control Loop

The flux dynamics depend on the rotor time constant $\tau_r = {}^{L_r}\!/_{R_r}$ and the magnetizing inductance $\hat{L}_m$, which is modified by the end-effect factor $f(Q)$.The gains are derived as:

$$\begin{cases} K_{p\phi} = \dfrac{2\zeta\omega_{n\phi}\tau_r - 1}{\hat{L}_m} \\ K_{i\phi} = \dfrac{\tau_r\omega_{n\phi}{}^2}{\hat{L}_m} \end{cases} \tag{21}$$

2.3.3. Outer Speed Loop

The mechanical tracking depends on the motor mass *M* and the friction coefficient *B*. The gains for speed regulation are:

$$\begin{cases} K_{p\omega} = 2\zeta\omega_{n\omega}M - B \\ K_{i\omega} = M\,\omega_{n\omega}{}^2 \end{cases} \tag{22}$$

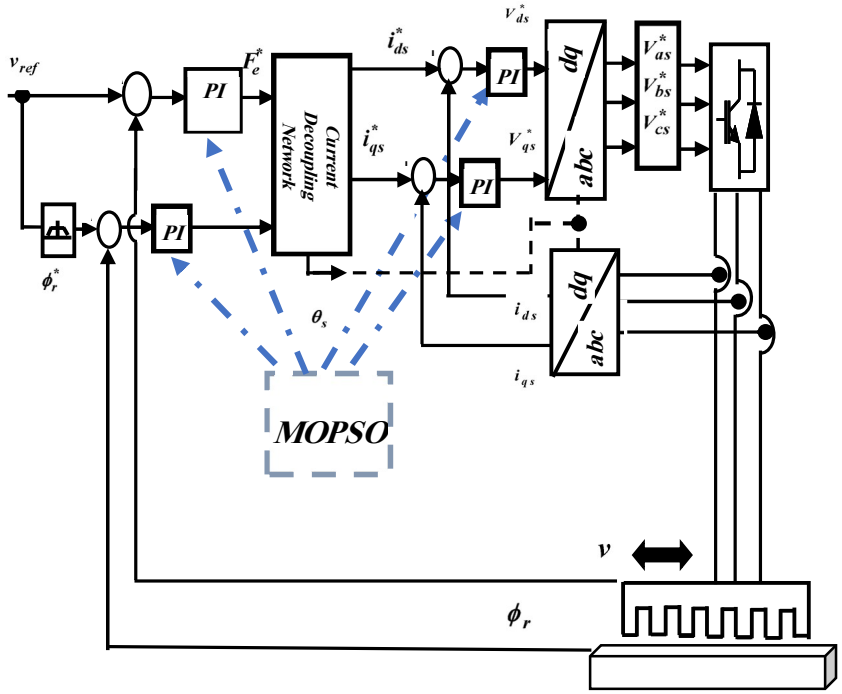


Figure 2

Detailed block diagram of MOPSO-FOC scheme for a LIM

3 *Optimization Methodology*

The complexity of optimizing multiple and conflicting objectives in the tuning of PI controllers for Field-Oriented Control (FOC) systems necessitates a robust and systematic approach. This section details the proposed methodology, which uses the MOPSO algorithm to effectively address this challenge. The optimization problem is formulated to simultaneously minimize speed tracking error, limit stator current magnitudes, and maintain the desired magnetic flux.

3.1. Particle Swarm Optimization (PSO)

Particle Swarm Optimization (PSO) is a population-based, stochastic optimization technique inspired by the collective intelligence of swarms [22] [23]. The algorithm operates on a population of candidate solutions, termed "particles," that move through a multi-dimensional search space as illustrated in the flowchart (Figure 3).

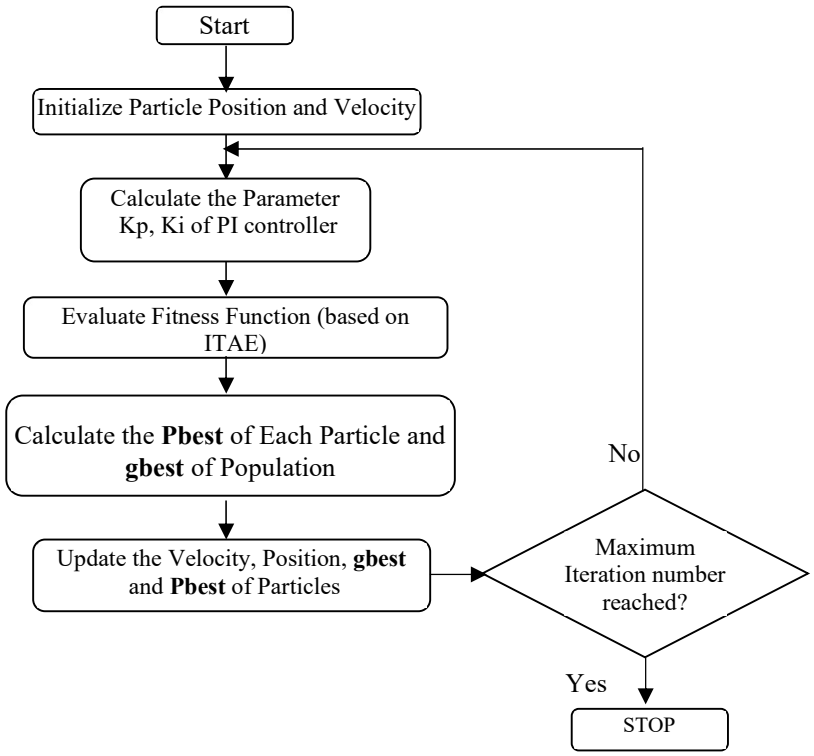


Figure 3

Flowchart for the PSO-PI Tuning Process

Each particle's movement is determined by its own experience and the experience of its neighbors. The state of each particle i at iteration t is defined by its position vector $\boldsymbol{x}_i\ (t)$ and its velocity vector $\boldsymbol{v}_i\ (t)$. The core of the algorithm is the update mechanism, where each particle's velocity is adjusted based on its best-known position, pbesti (personal best), and the best-known position of the entire swarm, gt (global best). The velocity and position update rules are as follows [24]:

$$\mathbf{v}_i(t+1) = \mathrm{w}\,.\mathbf{v}_i(t+1) + c_1.r_1.(\mathbf{pbest}_i(t) + \mathbf{x}_i(t)) + c_2.r_2.(\mathbf{gbest}_i(t) + \mathbf{x}_i(t)) \tag{23}$$

The position of each particle is updated in each generation according to following equation:

$$\mathbf{x}_i(t+1) = \mathbf{x}_i(t) + \mathbf{v}_i(t+1), \quad i = 1, 2, ...n \tag{24}$$

3.2. Multi-Objective PSO (MOPSO)

MOPSO is an extension of the classical Particle Swarm Optimization (PSO) algorithm designed to handle multiple objectives. It achieves this by replacing the

single global best solution with a set of non-dominated solutions that form the Pareto front [25].

A solution a is said to dominate another solution b if it is no worse than b across all objectives and strictly better in at least one:

$$f_i(\mathrm{a}) \le f_i(\mathrm{b}) \quad \forall\, \mathrm{j} \in \{1,....,M\}, \quad \text{and} \quad \exists k \ \text{ such that } \ f_k(\mathrm{a}) \le f_k(\mathrm{b}) \tag{25}$$

where M is the number of objectives. A solution is non-dominated if no other solution in the population dominates it. The set of all non-dominated solutions forms the Pareto front.

Furthermore, MOPSO maintains an external archive of non-dominated solutions found during the optimization. Leaders are selected from this archive to guide particles, with diversity preservation ensured through crowding distance or grid-based selection, preventing premature convergence and promoting uniform Pareto front distribution.

3.3. Problem Formulation

3.3.1. PI Controller Parameter Set

The proposed methodology focuses on tuning the PI controller gains for the four control loops of the LIM FOC system. The vector of parameters to be optimized is defined as:

$$\mathbf{x} = \left[\mathbf{K}_{pw}, \mathbf{K}_{iw}, \mathbf{K}_{p\phi}, \mathbf{K}_{i\phi}, \mathbf{K}_{pd}, \mathbf{K}_{id}, \mathbf{K}_{pq}, \mathbf{K}_{iq} \right] \tag{26}$$

Each particle in the MOPSO swarm represents a unique combination of these eight gains.

3.3.2. Objective Functions

The optimization problem is formulated to simultaneously minimize a set of objective functions [26]. The integral of time-weighted absolute error (ITAE) is chosen for each control loop as it provides a robust measure of both tracking performance and steady-state error. The objective vector **J** to be minimized is defined as:

$$\mathbf{J}(\mathbf{K}) = \begin{bmatrix} f_1(\mathbf{K}) \\ f_2(\mathbf{K}) \\ f_3(\mathbf{K}) \\ f_4(\mathbf{K}) \end{bmatrix} = \begin{bmatrix} ITAE_{\omega} \\ ITAE_{\phi} \\ ITAE_{id} \\ ITAE_{iq} \end{bmatrix} \tag{27}$$

Where K is the decision vector of PI gains:

$$\mathbf{K} = \left[\mathbf{K}_1, \mathbf{K}_2, \mathbf{K}_3, \mathbf{K}_4, \mathbf{K}_5, \mathbf{K}_6, \mathbf{K}_7, \mathbf{K}_8\right] \tag{28}$$

In this vector, the elements represent the proportional and integral gains of the PI controllers employed in the speed, flux, and current control loops of the drive system.

Specifically, the first two components of **K**, $\mathbf{K}_1$ and $\mathbf{K}_2$, correspond to the proportional and integral gains of the speed PI controller, i.e.,

$$\begin{cases} \mathbf{K}_1 = \mathbf{K}_{pw} \\ \mathbf{K}_2 = \mathbf{K}_{iw} \end{cases} \tag{29}$$

Similarly, the remaining components are defined as:

$$\begin{cases} \mathbf{K}_3 = \mathbf{K}_{p\phi}, \quad \mathbf{K}_4 = \mathbf{K}_{i\phi} \\ \mathbf{K}_5 = \mathbf{K}_{pd}, \quad \mathbf{K}_6 = \mathbf{K}_{id} \\ \mathbf{K}_7 = \mathbf{K}_{pq}, \quad \mathbf{K}_8 = \mathbf{K}_{iq} \end{cases} \tag{30}$$

where $\mathbf{K}_{p\phi}$ and $\mathbf{K}_{i\phi}$ denote the proportional and integral gains of the flux PI controller, while $\mathbf{K}_{pd}, \mathbf{K}_{id}$ and $\mathbf{K}_{pq}, \mathbf{K}_{iq}$ represent the gains of the d-axis and q-axis current PI controllers, respectively.

To ensure system stability and physical realizability, the search space is bound by:

$$\mathbf{K}_{j,\min} \le \mathbf{K}_j \le \mathbf{K}_{j,\max}, j = 1,....,8 \tag{31}$$

Each individual objective f_i is calculated using the Integral of Time-weighted Absolute Error:

- The first objective minimizes the ITAE of the speed error between the reference speed (w_{ref}) and the actual speed (w_{actual}) over the simulation time *tsim*.

$$f_1(\mathbf{K}) = \int_0^{tsim} \left| w_{ref} - w_{actual} \right| .t.dt \tag{32}$$

- The second objective aims to regulate the magnetic flux by minimizing the ITAE of the flux error.

$$f_2(\mathbf{K}) = \int_0^{tsim} \left| \phi_{ref} - \phi_{actual} \right| .t.dt \tag{33}$$

- The third and fourth objectives minimize the ITAE of the d- and q-axis stator currents, respectively.

$$\begin{cases} f_3(\mathbf{K}) = \int_0^{tsim} \left| ids_{ref} - ids_{actual} \right| .t.dt \\ f_4(\mathbf{K}) = \int_0^{tsim} \left| iqs_{ref} - iqs_{actual} \right| .t.dt \end{cases} \quad (34)$$

3.3.3. MOPSO Implementation Steps

The MOPSO algorithm solves this multi-objective problem through the following iterative procedure:

1. Initialization: Generate an initial swarm with random PI gains within predefined bounds and initialize velocities (Equation 28).
2. Simulation: For each particle, simulate the FOC-based LIM and calculate the four ITAE values (Equation 27).
3. Archive Update (Pareto Dominance): Non-dominated solutions are stored in an external archive. A solution K_1 dominates K_2 if it is no worse in all objectives and strictly better in at least one:

   $$\forall j \in \{1,...,4\} : f_j(\mathbf{k}_1) \le f_j(\mathbf{k}_2) \wedge \exists j \in \{1,...,4\} : f_j(\mathbf{k}_1) < f_j(\mathbf{k}_2) \quad (35)$$

   To maintain diversity, the Crowding Distance (CD) is calculated to prevent clustering:

   $$\mathbf{CD}_i = \sum_{m=1}^{4} \frac{f_m(i+1) - f_m(i-1)}{f_m^{\max} - f_m^{\min}} \quad (36)$$

   Solutions in less dense regions (higher CD) are prioritized to preserve population diversity.
4. Leader Selection: A "leader" is selected from the external archive using a density-based selection (e.g., roulette wheel based on CD) to guide particle movement toward unexplored optimal regions.
5. Velocity and Position Update: Update using MOPSO velocity equations with constraints applied to maintain parameter limits.
6. Mutation (Optional): Introduce random variations to avoid premature convergence.
7. Termination: Stop when the maximum number of iterations is reached or the Pareto front stabilizes.

4 Results and Discussions

In order to evaluate and validate the effectiveness of the proposed approach in FOC control of LIM with the consideration of the end-effects, two tests tracking and robustness were performed using the MATLAB/Simulink. In this research, the nominal parameters of LIM used for this simulation are as follows: Rs= 0.5 Ω, Rr= 0.3 Ω, Ls= 0.02 H, Lr= 0.015 H, Lm= 0.01 H, τ= 0.1 m. For implementing the (MOPSO) algorithm, several parameters are crucial for tuning the Proportional-Integral (PI) controller. A summary of these parameters can be found in Table.1 (Appendix).

4.1. Test Tracking

Furthermore, the simulations were running under dynamic operating conditions. These conditions included a step change in reference speed from 3 to 4 m/s at t = 4 sec and a sudden load force disturbance (FL) of 50 N applied at t = 1.0 sec. The following subsections analyze the key performance metrics, as well as a detailed comparison between the conventional PI controller and the MOPSO-tuned PI controller as presented in Table.2 (Appendix).

Figure 4 illustrates the actual speed response (blue line) compared to the reference speed (red line). The MOPSO-tuned controller demonstrates excellent tracking performance and closely following the reference speed with robust rejection of the load disturbance. Compared to the conventional PI controller tuned via the pole placement approach, the MOPSO approach achieves a faster rise time, reduced settling time, and minimized overshoot, with the steady state error reduced from 0.00875 to 0.00669. These improvements highlight the MOPSO approach enhanced transient response and steady-state accuracy.

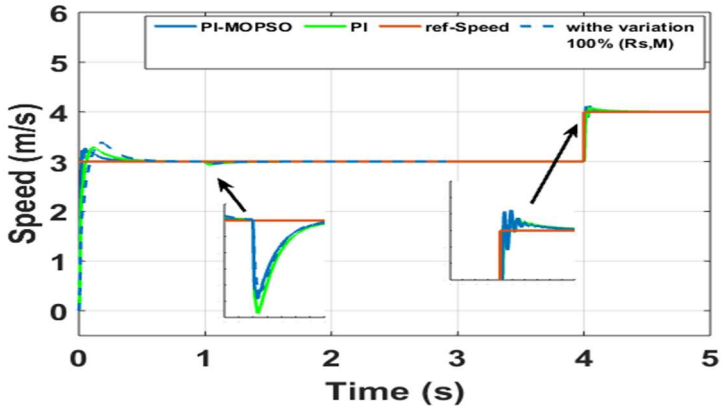


Figure 4

Speed response with PI conventional and MOPSO approach

As shown in Figure 5, the thrust force response under MOPSO optimization exhibits better tracking behavior. The optimized controller ensures a more stable and faster convergence to the desired thrust force with reduced overshoot and settling time compared to the conventional PI controller. This is critical for LIM applications that require precise linear motion control, where any oscillations or delays can compromise system efficiency and operational stability.

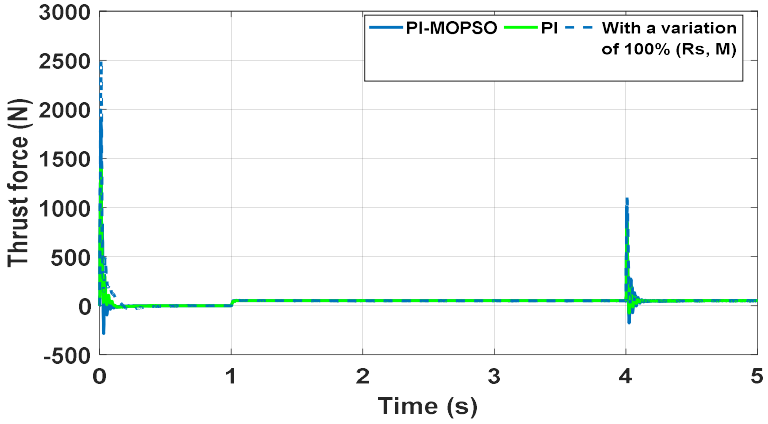


Figure 5

Thrust force response with PI-conventional and MOPSO approach

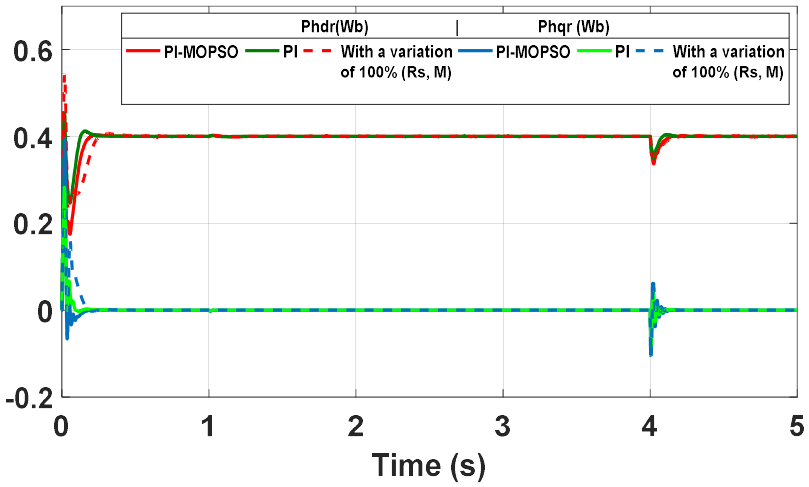
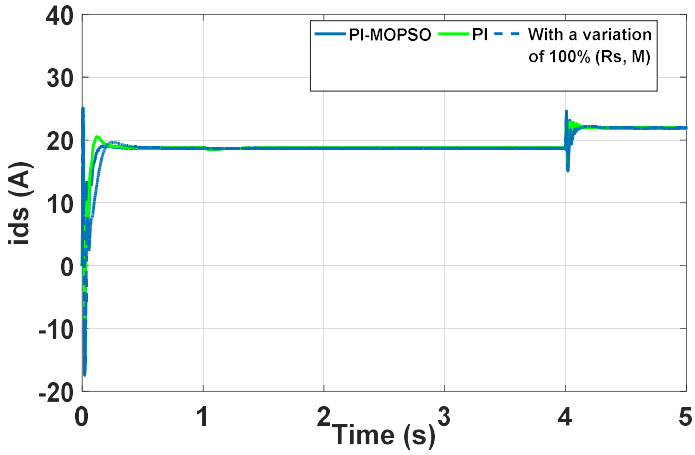


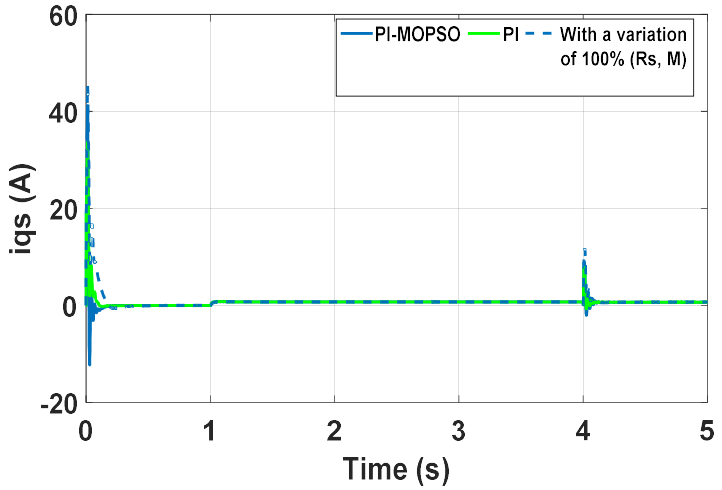
Figure 6

Direct and Quadrature flux response with PI conventional and MOPSO approach

Figure 6 presents the flux responses along the direct and quadrature axes. These results confirm that MOPSO contributes to better decoupling between the force and flux components, ensuring stable and accurate flux trajectories under dynamic conditions.



a)



b)

Figure 7

Direct and Quadrature current response with PI conventional and MOPSO approach

The direct (ids) and quadrature (iqs) current responses shown in Figure 7 demonstrate significant improvements in current regulation through the MOPSO

approach. Notably, the quadrature current (iqs), which directly influences the thrust force, exhibits enhanced tracking accuracy and stability. Similarly, the direct current (ids), responsible for flux regulation, also shows improved performance and contributes to the overall stability of the FOC system.

4.2. Robustness Test

To evaluate the robustness of the proposed approach, extensive parametric stress tests were conducted by simultaneously doubling the stator resistance Rs and the motor mass M. The dynamic response under these perturbed conditions remained stable and well-behaved despite the 100% increase in both parameters. The overshoot was effectively constrained to 12.63% for rotor speed, 33.43% for rotor flux magnitude, 43.5% for the d-axis current ids, and 44.0% for the q-axis current iqs. Although settling times increased to 0.34 s for speed and 0.40 s for iq, reflecting the greater challenge posed by the doubled parameters, the response exhibited markedly reduced oscillatory content during transients. This observation demonstrates excellent damping characteristics and transient stability. In steady state, the speed error was limited to 0.015, confirming that the MOPSO-optimized proportional and integral gains *Kp* and *Ki* provide robust tracking accuracy even under extreme parametric deviations. The comparative performance metrics summarized in Table 2 further highlight the superiority of the optimized controller over conventional methods. The proposed MOPSO approach maintained stable control characterized by faster settling times and minimal overshoot, successfully managing the inherent trade-off between responsiveness and damping.

Conclusions

This paper presents a novel approach to optimizing the Field-Oriented Control (FOC) strategy for Linear Induction Motors (LIMs), specifically addressing the challenges posed by end-effect phenomena. The primary contribution of this study lies in the application of the Multi-Objective Particle Swarm Optimization (MOPSO) technique to mitigate the drawbacks associated with the FOC technique, which arise from the conflicting tuning of proportional-integral (PI) controller gains across all control loops. The MOPSO algorithm was strategically employed to simultaneously and concurrently critical performance metrics, including speed tracking error, current regulation error, and magnetic flux deviation. Furthermore, the aim was to significantly enhance the transient response characteristics of the LIM system. This multi-objective process yielded a Pareto front of non-dominated solutions, effectively illustrating the optimal trade-offs achievable among the diverse performance objectives. Comprehensive simulation analyses, conducted under various dynamic operating conditions such as step changes in reference speed and sudden load disturbances, unequivocally demonstrated the superior performance of the MOPSO-tuned FOC. Comparative results highlighted substantial improvements over traditional tuning methodologies, notably exhibiting reduced overshoot by 3.98%, rise time by 56.34%, settling times by 38.13%, and steady-state error by 23.54%, as well as enhanced robustness against inherent system nonlinearities. These findings validate the efficacy and practical

applicability of the proposed MOPSO-based FOC approach, affirming its potential for achieving precise, stable, and energy-efficient control of LIM systems in industrial applications. Future work will focus on experimental validation and hardware-in-the-loop testing for real-time feasibility.

References

[1] MF. Elmorshedy, D. Almakhles, SM. Allam, "Improved performance of linear induction motors based on optimal duty cycle finite-set model predictive thrust control." Heliyon, Vol. 10, No. 13, 2024

[2] A. Accetta, M. Cirrincione, M. Pucci and A. Sferlazza, "State Space-Vector Model of Linear Induction Motors Including End-Effects and Iron Losses Part I: Theoretical Analysis," in IEEE Transactions on Industry Applications, Vol. 56, No. 1, Jan.-Feb. 2020, pp. 235-244

[3] L. Khettache, F. Djeffal and H. Ferhati, "Optimal Speed Control of Dual-star Linear Induction Motor based on Particle Swarm Algorithm," International Conference on Advances in Electrical and Communication Technologies (ICAECOT), Setif, Algeria, 2024, pp. 1-5

[4] H.Wang, Y.Yang, X.Ge, S.Li, & Y. Zuo, "Speed-Sensorless Control of Linear Induction Motor Based on the SSLKF-PLL Speed Estimation Scheme". I E E E Transactions on Industry Applications, Vol.56, No.5,2020. pp. 4986-5002

[5] A. Hentout, A. Maoudj, and A. Kouider, "Shortest path planning and efficient fuzzy logic control of mobile robots in indoor static and dynamic environments, " Romanian Journal of Information Science and Technology, vol. 27, no. 1, 2024, pp. 21–36

[6] D. Zhao., N. Zhao., H. Zhang., P. Shi, & I. Rudas, "Resilient sampled-data event-triggered control for switched systems under denial-of-service attacks, " Acta Polytechnica Hungarica, vol. 21, no. 10, 2024

[7] T. Haidegger., L. Kovács, R. E. Precup., B. Benyó., Z. Benyó., & S. Preitl., "Simulation and control for telerobots in space medicine, " Acta Astronautica, vol. 81, no. 1,2012, pp. 390–402

[8] L. Khettache., and R. Abdessemed. "A new speed control approach of linear induction motor based on robust RST controller and model reference adaptive system estimator." International Journal of Engineering, Vol.36, No.4, 2023, pp. 630-639

[9] C. Garcia, J. Rodriguez, C. Silva, C. Rojas, P. Zanchetta and H. Abu-Rub, "Full Predictive Cascaded Speed and Current Control of an Induction Machine," in IEEE Transactions on Energy Conversion, Vol. 31, No. 3, Sept. 2016, pp. 1059-1067

[10] L. Khettache., D. Djarah., G. Zidani., and R. Abdessemed., "Optimization of RST Controller for Speed Control of Linear Induction Motor Using Genetic

Algorithm." Recent Advances in Electrical & Electronic Engineering, Vol.17, No.3, 2024, pp. 260-268

[11] I. A. Zamfirache, R.-E. Precup, and E. M. Petriu, "Adaptive reinforcement learning-based control using proximal policy optimization and slime mould algorithm with experimental tower crane system validation, " Applied Soft Computing, vol. 160, Art. no. 111687, 2024

[12] M. I. Abdelwanis, "Linear Induction Motor Parameter Estimation Based on Gray Wolves Optimization Algorithm," 2022 23rd International Middle East Power Systems Conference (MEPCON), Cairo, Egypt, 2022, pp. 1-6

[13] D. Kumar, and K. Vijay. "Impact of controlling parameters on the performance of MOPSO algorithm." Procedia Computer Science, Vol. 167, 2020, pp. 2132-2139

[14] E. F. da Silva, E. B. dos Santos, P. C. M. Machado and M. A. A. de Oliveira, "Dynamic model for linear induction motors," IEEE International Conference on Industrial Technology, 2003, Maribor, Slovenia, Vol.1, 2003, pp. 478-482

[15] A. Majumdar and T. K. Bhattacharya, "Comparison of Force Developed in a Linear Induction Machine and an Equivalent Arc Linear Induction Machine at Zero Velocity," IEEE International Conference on Power Electronics, Drives and Energy Systems (PEDES), Chennai, India, 2018, pp. 1-5

[16] S. Rachev, K. Dimitrova, D. Koeva, and L. Dimitrov, "Starting Processes in Electric Drive of a Passenger Elevator", The Journal of CIEES, Vol. 1, No. 1, Jun.2021, pp. 40–49

[17] G. Lv, D. Zeng and T. Zhou, "An Advanced Equivalent Circuit Model for Linear Induction Motors," in IEEE Transactions on Industrial Electronics, Vol. 65, No. 9, Sept.2018, pp. 7495-7503

[18] F. Alonge, M. Cirrincione, M. Pucci, & A. Sferlazza, "Input–output feedback linearizing control of linear induction motor taking into consideration the end-effects. Part I: Theoretical analysis". Control Engineering Practice, Vol.36,2015, pp. 133-141

[19] K. Wang, Y. Li, Q. Ge and L. Shi, "An Improved Indirect Field-Oriented Control Scheme for Linear Induction Motor Traction Drives," in IEEE Transactions on Industrial Electronics, Vol. 65, No. 12, Dec. 2018, pp. 9928-9937

[20] S. Preitl and R.-E. Precup, "On the algorithmic design of a class of control systems based on providing the symmetry of open-loop Bode plots, " Scientific Bulletin of UPT, Transactions on Automatic Control and Computer Science, vol. 41, no. 55, 1996, pp. 47–55

[21] Á. Takács., L. Kovács, I. Rudas., R. E. Precup, & T. Haidegger., "Models for force control in telesurgical robot systems,"Acta Polytechnica Hungarica, vol. 12, no. 8,2015, pp. 95–114

[22] R.E. Precup, C. A. Bojan-Dragos., K. Gao., & S. Cui., "Problem setting for trajectory planning and cruise control of a connected autonomous electric bus in intersection scenarios with human-driven vehicles to optimize energy, comfort and tracking, " Romanian Journal of Information Science and Technology, vol. 28, 2025, pp. 299–312

[23] D. Zellouma, Y. Bekakra, and H. Benbouhenni. "Field-oriented control based on parallel proportional–integral controllers of induction motor drive." Energy Reports, Vol. 9, 2023, pp. 4846-4860

[24] G. LI and W. CHOU, "Path planning for mobile robot using self-adaptive learning particle swarm optimization", Science China Information Sciences 61(5), 2018, paper 052204

[25] U. Nagalingam, B. Mahadevan, K. Vijayarajan, and A. P. Loganathan, "Design optimization for cogging torque mitigation in brushless DC motor using multi-objective particle swarm optimization algorithm, " COMPEL – The International Journal for Computation and Mathematics in Electrical and Electronic Engineering, vol. 34, no. 4, 2015, pp. 1302–1318

[26] R. H. A. Hamid, A. M. A. Amin, R. S. Ahmed and A. A. A. El-Gammal, "Optimal Operation of Induction Motors Based on Multi-Objective Particle Swarm Optimization (MOPSO)," IECON 2007 - 33rd Annual Conference of the IEEE Industrial Electronics Society, Taipei, Taiwan, 2007, pp. 1079-1084

Appendix

Table 1

Parameters Used in the MOPSO Algorithm

Parameter	Description	Value
Population size	Number of particles in the swarm	20
wmin ,wmax	Initial and Final inertia weight	0.1 , 0.9
C1, C2	Cognitive and Social acceleration factor	1.5
Itermax	Maximum number of iterations	50
M	Control parameter for chaotic sequences	4.0
F	Initial value (random in [0,1])	Random [0,1]

Table 2

Comparative Analysis of Performance Metrics and Controller Gains: Conventional PI vs. PI-MOPSO

Method	Performance criteria	Speed *v*	Currant *ids*	Currant *iqs*	Flux *ϕdr*	Flux *ϕqr*
PI	Overshoot %	9.4514	22.9133	41.087	13.6011	39.5
	SettlingTime	0.3362	0.2012	0.0944	0.1840	0.858
	RiseTime	0.0339	0.0028	7.56e-06	0.0073	8.01e-06
	Error	0.0087	0.00134	0.00116	1.09e-04	3.37e-06
	Kp	6.48	9.15	9.14	6.93	/
	Ki	44.32	6.7e+03	6.5e+03	1.88e+03	/
PI-MOPSO	Overshoot %	9.075	34.22	33.60	12.55	28.26
	SettlingTime	0.208	0.118	0.011	0.1718	0.01343
	RiseTime	0.0148	0.00237	1.3e-07	0.0069	2.9e-05
	Error	0.0066	1.7e-05	4.17e-04	7.96e-05	1.72e-06
	Kp	7.81	8.97	8.09	6.918	/
	Ki	55	6792	6387	996.63	/
PI-MOPSO With a variation of 100% (Rs M)	Overshoot %	12.63	33.43	43.5	15.6	44
	SettlingTime	0.34	0.38	0.4	0.37	0.138
	RiseTime	0.5	0.024	0.07	0.25	0.21
	Error	0.015	0.03	0.08	0.011	6e-03