Abstract—In this paper, discrete-time inverse optimal control for trajectory tracking is applied for a three-phase linear induction motor (LIM). The system mathematical model is in the Nonlinear Block Controller Form (NBCF) in order to achieve trajectory tracking. The control law computes the input voltage signals which are optimal in the sense that this control law minimizes a meaningful cost functional. A Particle Swarm Optimization (PSO) algorithm is employed to determine the matrix $P$ for inverse optimal control in order to improve tracking results. Simulation illustrate the applicability of the proposed control scheme.

I. INTRODUCTION

Linear electric actuators are electromechanical devices which produce unidirectional or bidirectional short-stroke motion [1]. The linear induction motor (LIM) is a linear electric actuator on which the electrical energy is turned into mechanical translational movement; this is, a mobile element on the motor moves linearly with respect to a stationary element [2].

LIMs present several advantages with respect to other types of motors. The LIM develops magnetic forces directly between the mobile element and the stationary element, without the need of physical contact between both elements, which would restrict the system dynamics. Then, the LIM can reach higher speed and reduces undesirables vibrations [3].

For this reason the LIM has been employed in a wide sector of the industry as in the steel, textile, nuclear and spatial industries [4]. Nevertheless, the most extensive application for LIMs is for transportation, where high speed trains have been constructed using the rails as the stationary element and with the mobile element of the motor fixed in the train.

The idea of using linear motors for mass public transportation is not new [5]. However, the attention is focused on the LIM again due to recent developments of large smart grids. The modern electric power grid with wind energy is a complex adaptive system with the presence of unknown uncertainties and disturbances [6]. The integration of plug-in hybrid and electric vehicles causes that the control problem of power grids can be very difficult to handle with traditional approaches, requiring the application of intelligent control approaches [7].

Optimal control is related to find a control law for a given system such that a performance criterion, usually formulated as a cost functional, is minimized [8]. The optimal control problem can be stated as follows: given a meaningful cost functional we have to find the control law that minimizes it [8],[9]. Optimal control can be solved using Pontryagin maximum principle [10], or dynamic programing developed by Bellman [11], [12], which leads to a nonlinear partial differential equation called the Hamilton-Jacobi-Bellman (HJB) equation; Solving this equation is not straightforward: for systems of dimension higher than two there are no practical ways to solve the HJB equation [9],[13].

Inverse optimal control is a solution for optimal control, which avoid the need to solve the associated HJB partial differential equation [8], [9]. For the inverse approach, a stabilizing feedback control law, based on a priori knowledge of a Control Lyapunov Function (CLF), is designed first and then it is established that this control law optimizes a meaningful cost functional which is a posteriori determined.

Evolutionary computation techniques (evolutionary programming, genetic algorithms and evolutionary strategies) are motivated by the evolution of nature. A population of individuals, which encode the problem solutions, are manipulated according to some rules which evolve in a better solution through the generations [14]. Particle Swarm Optimization (PSO) is an algorithm which emulates social behavior. Birds flocks, fish schools and animal herds constitute representative examples of natural systems where aggregated behaviors are met, producing impressive, collision-free, synchronized moves. For such systems, the behavior of each group member is based on simple inherent responses [15]. In PSO each particle adjusts its flying according to its own experience and its companions one; each particle represents a potential solution of the optimization problem and travels through the search space for the best solution. For inverse optimal control theory, the method to find an appropriate matrix $P$ is traditionally heuristic. In this paper employment of PSO algorithm is proposed in order to get better results.

The reminder of this paper is organized as follows: in section II the mathematical model of the LIM is presented; in section III the full controller scheme is described; in section IV two cases are presented: inverse optimal control (case 1) and PSO inverse optimal control (case 2) is applied to the LIM.
model; simulation results are included. Finally, in section V conclusions about the obtained results and proposed future work are given.

II. DISCRETE-TIME MODEL OF THE LINEAR INDUCTION MOTOR

The system mathematical model employed in this paper is described by the discrete-time α − β model of the linear induction motor [2],[16]. This model does not take into account the end effects [17],[18] which can be despised depending on the dimensions of the LIM. The system model employed in this paper is described by the following equations:

\[ q_{k+1} = q_k + v_k T \]

\[ v_{k+1} = (1 - k_2 T) v_k - k_3 T F_L - k_1 T \lambda_{\alpha,k} q_k + v_k T \lambda_{\alpha,k} q_k \]

\[ -k_1 T \lambda_{\beta,k} m_{\alpha,k} + k_1 T \lambda_{\alpha,k} m_{\beta,k} - k_1 T \lambda_{\alpha,k} q_k \]

\[ \lambda_{\alpha,k} = (1 - k_6 T) \lambda_{\alpha,k} + k_4 T v_k p_1 i_{\alpha,k} - k_4 T v_k p_1 i_{\alpha,k} \]

\[ + k_5 T p_2 i_{\alpha,k} - k_4 T v_k p_2 i_{\beta,k} + k_5 T p_2 i_{\beta,k} \]

\[ \lambda_{\beta,k} = (1 - k_6 T) \lambda_{\beta,k} + k_4 T v_k p_2 i_{\beta,k} - k_4 T v_k p_2 i_{\alpha,k} \]

\[ + k_5 T p_1 i_{\beta,k} - k_4 T v_k p_1 i_{\beta,k} + k_5 T p_1 i_{\alpha,k} \]

\[ i_{\alpha,k} = (1 + k_9 T) i_{\alpha,k} - k_7 T \lambda_{\alpha,k} p_2 - k_8 T \lambda_{\alpha,k} v_k p_1 \]

\[ + k_7 T \lambda_{\alpha,k} p_1 - k_8 T \lambda_{\alpha,k} v_k p_2 - k_10 T u_{\alpha,k} \]

\[ i_{\beta,k} = (1 + k_9 T) i_{\beta,k} + k_7 T \lambda_{\beta,k} p_2 - k_8 T \lambda_{\beta,k} v_k p_1 \]

\[ - k_7 T \lambda_{\beta,k} p_1 - k_8 T \lambda_{\beta,k} v_k p_2 - k_10 T u_{\beta,k} \]

where

\[ \rho_1 = \sin(n_p q(k)) \quad \rho_2 = \cos(n_p q(k)) \]

\[ k_1 = \frac{n_p L_{sr}}{D_m L_r} \quad k_2 = \frac{R_m}{D_m} \quad k_3 = \frac{1}{D_m} \]

\[ k_4 = \frac{n_p L_{sr}}{R_m} \quad k_5 = \frac{L_{sr}}{L_r} \quad k_6 = \frac{R_m}{L_r} \]

\[ k_7 = \frac{L_{sr} R_r}{L_r (L_{sr}^2 - L_{sr} L_r)} \quad k_8 = \frac{L_{sr} n_p}{L_{sr}^2 - L_{sr} L_r} \]

\[ k_9 = \frac{L_{sr} R_r}{L_r (L_{sr}^2 - L_{sr} L_r)} \quad k_{10} = \frac{L_r}{L_{sr}^2 - L_{sr} L_r} \]

A. Inverse Optimal Control

Consider the affine discrete-time nonlinear system

\[ x_{k+1} = f(x_k) + g(x_k) u_k \]

and the proposed CLF as

\[ V(x_k) = \frac{1}{2} x_k^T P x_k, \quad P = P^T > 0 \]

Then the inverse optimal control law takes the following form

\[ u_k^* = -\frac{1}{2} (R(x_k) + P_2(x_k))^{-1} P_1(x_k) \]

where \( P_1(x_k) = g^T(x_k) P f(x_k) \) and \( P_2 = \frac{1}{2} g^T(x_k) P g(x_k) \)

Theorem 1: [8] Consider the system (3). If there exist a matrix \( P = P^T > 0 \) such that the following inequality holds:

\[ V_f(x_k) - \frac{1}{4} P_1^T(x_k) (R(x_k) + P_2(x_k))^{-1} P_1(x_k) \leq -\varsigma Q \| x_k \|^2 \]

where \( V_f(x_k) = \frac{1}{2} [ f^T(x_k) P f(x_k) - V(x_k) ] \) and \( \varsigma Q > 0 \). \( P_1 \) and \( P_2 \) as defined in (5); then the equilibrium point \( x_k = 0 \) of system (3) is globally exponentially stabilized by the control law (5), with CLF (4). Moreover, this control law is inverse optimal in the sense that it minimizes the meaningful functional given by.

\[ J(x_k) = \sum_{k=0}^{\infty} (l(x_k) + u_k^T R(x_k) u_k) \]

with \( l(x_k) = -V(x_{k+1}) + V(x_k) - u_k^T R(x_k) u_k^* \).

For the detailed proof of Theorem 1 we refer the reader to [8].

B. Nonlinear Block Controllable Form

The block control method is applied to decompose the control law synthesis problem into a number of sub-problems of lower order. A special state representation of system (3) is used, which is referred as the Nonlinear Block Controllable (NBC) form consisting of \( r \) blocks [19]:

\[ x_{k+1} = f^1(x_k) + B^1(x_k) x_k^2 \]

\[ x_{k+1}^2 = f^2(x_k, x_k^2) + B^2(x_k, x_k^2) x_k^3 \]

\[ x_{k+r}^r = f^r(x_k, x_k^2, \ldots, x_k^{r-1}) + B^{r-1}(x_k, x_k^2, \ldots, x_k^{r-1}) x_k^r \]

where \( x_k \in \mathbb{R}^n \), \( x_k = [x_k^T x_k^2 \ldots x_k^r]^T \), \( x_j \in \mathbb{R}^{n_j} \); \( j = 1, \ldots, r \); \( n_j \) denotes the order of each rth block; \( x_j = [x_j^1 x_j^2 \ldots x_j^{n_j}] \); input \( \alpha(x_k) \in \mathbb{R}^n \).

For trajectory tracking of first block in (8), let define the tracking error as

\[ z_{k+1}^1 = x_{k+1}^1 - x_{k+1}^1 \]

where \( x_{k+1}^1 \) is the desired trajectory signal. Taking one step ahead in (9) we have

\[ z_{k+1}^1 = f^1(x_k^1) + B^1(x_k^1) x_k^2 - x_{k+1}^1 \]

III. CONTROL SCHEME
Equation (10) is viewed as a block with state $z^1_k$ and the state $x^2_k$ is considered as a pseudo-control input where desired dynamics can be imposed as follows:

$$z^1_{k+1} = f^1(x^1_k) + B^1(x^1_k)x^2_k - x^3_{k+1} + K_1z^1_k$$  \hspace{1cm} (11)$$

where $K_1 = \text{diag}\{k_{11}, ..., k_{n1}\}$ is a Schur matrix such that $K_1z^1_k$ is a stable dynamic. Then, the desired behavior of $x^2_k$ is calculated as

$$x^2_{k,k} = (B^1(x^1_k))^{-1}(x^1_{k,k+1} - f^1(x^1_k) + K_1z^1_k)$$  \hspace{1cm} (12)$$

Proceeding along the same way as for the first block, a second variable in the new coordinates is defined as

$$z^2_k = x^2_k - x^3_{k,k}$$  \hspace{1cm} (13)$$

and the desired behavior for $x^3_k$ can be calculated. Taking these steps iteratively, the last new variable is defined as

$$z^3_k = x^3_k - x^3_{k,k}$$  \hspace{1cm} (14)$$

Taking one step ahead yields

$$z^3_{k+1} = f^r(x_k) + B^r(x_k)\alpha(x_k) - x^3_{k,k+1}$$  \hspace{1cm} (15)$$

By means of this change of variables, system (8) can be represented as

$$z^1_{k+1} = K_1z^1_k + B^1(x^1_k)z^2_k$$

$$z^2_{k+1} = K_2z^2_k + B^2(x^1_k,x^2_k)z^3_k$$

$$\vdots$$

$$z^3_{k+1} = f^r(x_k) - x^3_{k,k+1} + B^r(x_k)\alpha(x_k)$$  \hspace{1cm} (16)$$

Now when the new error variables tend to zero, $x^1_k$ will tend to $x^1_{k,k}$ as desired. Then, a stabilizing control law $\alpha(x_k)$ can be used to achieve trajectory tracking.

C. Particle Swarm Optimization

In PSO algorithm, each member of the population is treated as a point in a D-dimensional space. The ith particle is represented as $X_i = (x_{i1}, x_{i2}, ..., x_{iD})$. The best previous position of each particle is recorded and represented as $P_i = (p_{i1}, p_{i2}, ..., p_{iD})$. The best particle among all the particles (the global best position) is represented by $p_{gd}$. The rate of the position change, or velocity of particle $i$ is represented as $V_i = (v_{i1}, v_{i2}, ..., v_{iD})$. Then, the particles are manipulated according to the following equation:

$$v_{id} = v_{id} + c_1r_1d_1(p_{id} - x_{id}) + r_2d_2(p_{gd} - x_{id})$$  \hspace{1cm} (17)$$

$$x_{id} = x_{id} + v_{id}$$  \hspace{1cm} (18)$$

where $c_1$ and $c_2$ are two positive constants, $r_1d_1$ and $r_2d_2$ are two random functions in the range [0, 1]. The performance of each particle is measured according to a predefined fitness function, which is related to the problem to be solved.

In this paper, PSO is employed to find the matrix $P$ for inverse optimal control. The algorithm computes the matrix which minimizes the mean square tracking error for the motor position with respect to the given trajectory reference.

IV. SIMULATION RESULTS

In this section, inverse optimal trajectory tracking is applied to the discrete-time model of the LIM (1)-(2). Fig. 1 is a block diagram which represents how the control law and the system states minimize a meaningful cost functional. The control law is calculated by means of a state feedback; for this reason the full state of the system is supposed to be available. The LIM model is organized in the NBC form to achieve trajectory tracking for the position of the secondary sector of the motor.

A. Linear Induction Motor Block Controller Form

In general, electromechanical systems are or can be easily written in the NBC form. In order to organize the LIM model in this form, it is necessary to reduce the system order. We achieved this goal by taking the magnetic flows magnitude instead of each magnetic flow separately, that is

$$\psi_{k+1} = \lambda^2_\alpha_{k+1} + \lambda^2_\beta_{k+1}$$  \hspace{1cm} (19)$$

Then, the LIM model can be separated in three different blocks:

$$x^1_k = q_k, \quad x^2_k = \begin{bmatrix} v_k \\ \psi_k \end{bmatrix}, \quad x^3_k = \begin{bmatrix} i_{\alpha,k} \\ i_{\beta,k} \end{bmatrix}$$  \hspace{1cm} (20)$$

A change of variables can be done as in (16) so that the system with desired dynamics is represented by:

$$z^1_{k+1} = K_1z^1_k + B^1z^2_k$$

$$z^2_{k+1} = K_2z^2_k + B^2(x^1_k,x^2_k)z^3_k$$

$$z^3_{k+1} = f^3(x_k) - x^3_{k,k+1} + B^r(x_k)\alpha(x_k)$$  \hspace{1cm} (21)$$

where $B^1, B^2(x_k), f^3(x_k)$ and $B^3$ are the corresponding functions obtained from the system (1). Trajectory tracking can now be achieved for the position reference $x^1_{k,k}$.

B. Control Law Implementation

In (21) $z^1_k$ and $z^2_k$ will have stable dynamics if the third state block $z^3_k$ tends to zero when $k \to \infty$. Then, a control law has to be designed such that it stabilizes $z^3_k$. The inverse optimal control law applied to the system is of the following form:

$$u_k = -\frac{1}{2}(R + \frac{1}{2}B^{3T}PB^3)^{-1}B^{3T}Pf^3(x_k)$$  \hspace{1cm} (22)$$

where $R = I_2$, $I_2$ is the $2 \times 2$ identity matrix, $B^3$ and $f^3(x_k)$ are defined in (21) and $T$ denotes the transpose matrix.
The reference signal for the motor position is selected as

\[ x_{1,k}^1 = 0.25 \sin(\pi t) + 0.25 \]  \hspace{1cm} (23)

Case 1. Initially, a matrix \( P \) is selected such that control law (22) achieves asymptotical stabilization for the system. The selected matrix \( P \) is

\[ P = \begin{bmatrix} 2 & 0 \\ 0 & 2 \end{bmatrix} \]  \hspace{1cm} (24)

The numerical values of the system parameters employed in simulation are specified in Table I. The sampling time can be increased; however, it must be carefully selected. The upper limit of the sampling time \( T \) is bounded by two reasons: first, the lost of information in the sampled signal with a large \( T \); second, the influence of the sampling time on the closed loop stability since it increases the range of instability of the system. Simulation results are displayed in Fig.2 where the dotted line is the reference trajectory signal and the continuous line is the system output signal.

Case 2. As mentioned above, PSO algorithm is employed in order to find the matrix \( P \). Figure 3 shows a block diagram which represents the control scheme where PSO affects the control law by means of the matrix \( P \). The output matrix of the PSO algorithm is:

\[ P = \begin{bmatrix} 56.16 & -5.31 \\ -5.31 & 52.14 \end{bmatrix} \]  \hspace{1cm} (25)

which is symmetric and positive definite as needed by Theorem 1.

Simulation with the new matrix \( P \) is done for the same conditions as the one in the previous case; the results are shown in Figure 4.

For this case, trajectory tracking is achieved in 0.225 seconds with a mean square error of 0.00109. In the previous case, trajectory tracking is achieved in 0.718 seconds with a mean square error of 0.0048. The difference between both results shows the better performance of the proposed PSO-based control law.

In order to test this control scheme with a different reference signal, the following step train signal is applied

\[ x_{1,k}^1 = 0.25U_k + 0.3U_{k-1} - \frac{U_{k-2}}{2} + 0.2U_{k-3} - \frac{U_{k-4}}{2} + 0.1U_{k-5} - \frac{U_{k-6}}{2} - 0.3U_{k-7} - \frac{U_{k-8}}{2} - 0.3U_{k-9} - \frac{U_{k-10}}{2} \]  \hspace{1cm} (26)

where \( U_k \) is the unit step function, defined as

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>( R_s )</td>
<td>5.3685 Ω</td>
</tr>
<tr>
<td>( R_r )</td>
<td>3.5315 Ω</td>
</tr>
<tr>
<td>( L_s )</td>
<td>28.46 mH</td>
</tr>
<tr>
<td>( L_r )</td>
<td>28.46 mH</td>
</tr>
<tr>
<td>( L_m )</td>
<td>24.1 mH</td>
</tr>
<tr>
<td>( n_p )</td>
<td>4</td>
</tr>
<tr>
<td>( R_m )</td>
<td>36.0455 Kg/s</td>
</tr>
<tr>
<td>( D_m )</td>
<td>2.78 Kg</td>
</tr>
<tr>
<td>( K_f )</td>
<td>148.38 N/A</td>
</tr>
<tr>
<td>( T )</td>
<td>0.0001 s</td>
</tr>
</tbody>
</table>

Table I
Simulation values for the model parameters
Position Trajectory Tracking

Figure 5. PSO improved trajectory tracking for the motor position

\[
U_k = \begin{cases} 
0, & \text{if } k < 0 \\
1, & \text{if } k \geq 0 
\end{cases}
\]  

(27)

Simulation result for this new reference signal and the described PSO, is displayed in Figure 5. The tracking mean square error is 0.0013531.

V. CONCLUSIONS

An inverse optimal controller for nonlinear discrete-time systems is applied to an affine model representing LIM dynamics. The proposed scheme achieves tracking for two different position trajectory references. PSO algorithm improves trajectory tracking. Indeed, convergence rate of the control law is increased. Simulation results show the difference between the control law performance before and after the PSO implementation. Considering the successful results, neural identification and real-time application are proposed as future work.

REFERENCES