

Article

Germinal Center Optimization Applied to Neural Inverse Optimal Control for an All-Terrain Tracked Robot

Carlos Villaseñor ¹, Jorge D. Rios ¹, Nancy Arana-Daniel ^{1,*} , Alma Y. Alanis ¹ ,
Carlos Lopez-Franco ¹ and Esteban A. Hernandez-Vargas ²

¹ Centro Universitario de Ciencias Exactas e Ingenierías, Universidad de Guadalajara, Blvd. Marcelino García Barragán 1421, 44430 Guadalajara, Mexico; cavp@outlook.com (C.V.); jorge_rios.1xyz@yahoo.com (J.D.R.); almayalanis@gmail.com (A.Y.A.); clzfranco@gmail.com (C.L.-F.)

² Frankfurt Institute For Advanced Studies, Ruth-Moufang-Straße 1, 60438 Frankfurt am Main, Germany; vargas@fias.uni-frankfurt.de

* Correspondence: nancy.arana@cucei.udg.mx; Tel.: +52-331-547-3877

Received: 16 November 2017; Accepted: 21 December 2017; Published: 27 December 2017

Abstract: Nowadays, there are several meta-heuristics algorithms which offer solutions for multi-variate optimization problems. These algorithms use a population of candidate solutions which explore the search space, where the leadership plays a big role in the exploration-exploitation equilibrium. In this work, we propose to use a Germinal Center Optimization algorithm (GCO) which implements temporal leadership through modeling a non-uniform competitive-based distribution for particle selection. GCO is used to find an optimal set of parameters for a neural inverse optimal control applied to all-terrain tracked robot. In the Neural Inverse Optimal Control (NIOC) scheme, a neural identifier, based on Recurrent High Order Neural Network (RHONN) trained with an extended kalman filter algorithm, is used to obtain a model of the system, then, a control law is design using such model with the inverse optimal control approach. The RHONN identifier is developed without knowledge of the plant model or its parameters, on the other hand, the inverse optimal control is designed for tracking velocity references. Applicability of the proposed scheme is illustrated using simulations results as well as real-time experimental results with an all-terrain tracked robot.

Keywords: Germinal Center Optimization; Artificial Immune Systems; Evolutionary Computing; neural identification; inverse optimal control; extended kalman filter

1. Introduction

Nowadays, in computer science research is important to offer optimal techniques for a variety of problems, nevertheless, for most of these problems are difficult to formalize a mathematical model to optimize. Soft-computing optimization techniques, such as Evolutionary Computing (EC) [1], Artificial Neural Networks (ANN) [2] and Artificial Immune Systems (AIS) [3–5], approach these kinds of problems by offering good approximate solutions in an affordable time. EC algorithms offer an analogy of the competitive process in natural selection applied to multi-agent search for multi-variate problems, in the same way, AIS are based on the adaptive properties of the vertebrates immune system.

The vertebrates immune system has been developed through time by natural selection to overcome many diseases, although some of this protection mechanisms are inheritable, the immune system is capable of adapting to a new variety of Antigens (AGs) (foreign particles) in order to acquire specific protection [6]. This specific protection is given by Antibodies (ABs) that attach to AGs with certain affinity in the so-called humoral immunity. ABs are produced by the differentiation of the lymphocyte

B (B-cell). In the case that the body does not have a specific AB for an AG, the B-cells compete for producing a better affinity AB with the help of lymphocyte T CD4⁺ (Th-cell), this competition is the inspiration of the Clonal Selection algorithm [7], which has many variants and improvements [8,9].

When an infection prevails, the innate immune response is not capable of managing it. In this case, the adaptive immune response starts a process called clonal expansion of B-cells, looking for a B-cell with high-affinity ABs [6]. The highest affinity of ABs is achieved by a biological process called Germinal Center reaction. The Germinal Centers are temporal sites in the secondary lymph nodes histologically recognizable, where inactive B-cell enclose active B-cells, Follicular Dendritic Cells (FDC) and Th-cells with the objective of maturing the affinity through a competitive process. For a better understanding of the biological phenomenon, we refer the interested readers to [10].

In this paper, we use Germinal Center Optimization (GCO), a new multi-variate optimization algorithm, inspired by the germinal center reaction, that hybridizes some concepts of EC and AIS, for optimization of an inverse optimal controller applied to an all-terrain tracked robot. The principal feature of GCO is that the particle selection for crossover is guided by a competitive-based non-uniform distribution, this embedded the idea of temporal leadership, as we explain in Section 2.3.

On the other hand, most of the modern control techniques need the knowledge of a mathematical model of the system to be controlled. This model can be obtained using system identification in which the model is obtained using a set of data obtained from practical experiments with the system. Even when the system identification technique does not obtain an exact model, satisfactory models can be obtained with reasonable effort. There is a number of system identification techniques, to name a few: neural networks, fuzzy logic, auxiliary model, hierarchical identification. Among these system identification techniques, system identification using neural networks stands out, especially using recurrent neural networks which have a dynamic behavior [2,11,12].

The Recurrent High Order Neural Networks (RHONNs) are a generalization of the first order Hopfield network [11,12]. The presence of recurrent and high order connections gives the RHONN compared to a first order feedforward neural networks [11–13]: strong approximation capabilities, a faster convergence, greater storage capacity, a Higher fault tolerance, robustness against noise and dynamic behavior. Also, the RHONNs have the following characteristics [11,12,14]:

- They allow an efficient modeling of complex dynamic systems, even those with time-delays.
- They are good candidates for identification, state estimation, and control.
- Easy implementation.
- A priori information of the system to be identified can be added to the RHONN model.
- On-line or off-line training is possible.

The goal of the inverse optimal control is to determine a control law which forces the system to satisfy some restrictions and at the same time to minimize a cost functional. The difference with the optimal control methodology is that the inverse optimal control avoids the need of solving the associated Hamilton-Jacobi-Bellman (HJB) equation which is not an easy task and it has not been solved for general nonlinear systems. Furthermore, 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 cost functional [15].

The control scheme consisting of a neural identifier and an inverse optimal control technique is named neural inverse optimal control (NIOC), this control scheme has shown good results in the literature for trajectory tracking [15–17]. However, the designer has to tune the appropriate value of some parameters of the controller discuss later in this work, the quality of the controller depends directly on this selection.

In this work, the main contribution is the introduction of an optimization process using GCO, in order to find the appropriate values for the controller parameters which minimize the tracking error of the system to be controlled. Performance of the optimization is shown presenting simulation and experimental tests comparing the results of the trajectory tracking using the NIOC with the parameters selected by the designer and the results using the parameters given by the GCO algorithm.

This work is organized as follows: In Section 2 the Germinal Center Optimization algorithm is described. In Section 2.1 the vertebrates adaptive immune system is briefly explained, in Section 2.2 we detail the germinal center reaction and in Section 2.3 the computation analogy of the germinal center reaction is presented along with the algorithm description. Section 3 introduces the Neural Inverse Optimal Control (NIOC) scheme for this work, where Section 3.1 presents the RHONN identifier and the extended kalman filter (EKF) training, and in Section 3.1.2 the design of the inverse optimal control law is discussed. Section 4 unveils comparative simulations (Section 4.2) and experimental (Section 4.3) results between the selection of the parameter of the controller using the GCO algorithm and the classic way which is let completely to the designer for an application of the NIOC to an All-Terrain Tracked Robot. Conclusions of this work are included in Section 5.

2. Germinal Center Optimization

In this section, we briefly overview the principal processes in the vertebrates immune system, and we detail the germinal center reaction. After that, we propose the computational analogy for multi-variate optimization, with the proper algorithm description.

2.1. Adaptive Immune System

The vertebrates immune system (VIS) is the biological mechanism for protecting the body from AG. There are two types of immunities, the innate immunity, and the adaptive one. The innate immunity is conformed by epithelial barriers that prevent the entrance of AG, phagocytes that swallow AG, FDCs that capture antigen and lymphocytes NKs (Natural Killers) that destroy any non-self cell.

The innate immunity is an inheritable protection that has been developed through natural selection, but if a new type of AG gets inside the body could overcome this basic protections, in this case, the adaptive immunity takes place. The adaptive immunity is conformed by B-cells whose main functions are to internalize AG for presentation and generate ABs, Th-cells that give a life signal to high-affinity B-cells and cytotoxic T-cells that kills own cells that are already infected or kills carcinogenic cells [6].

The affinity of the innate immunity is not diverse because it is coded in the germinal line, in the other hand the adaptive immunity has a high-affinity diversity because the receptors are produced for somatic recombination and variate with somatic hyper-mutation [6,18].

There are two types of adaptive immune response, the humoral immunity, and the cellular immunity. The first one is based on ABs that travel in the bloodstream, attaching to every compatible particle. The B-cells compete for antigen internalization and presentation to the Th-cell, whose reward them with a life signal, then the B-cell proliferates by clonal expansion and differentiates into plasmatic cells, releasing higher affinity ABs in the bloodstream.

There are some AGs that infect the owner cells and hide inside them, these cells are destroyed by the cellular immunity with the cytotoxic T-cell. The adaptive immune response has the following features, as is shown in [6]:

- Specificity: Ensure to produce a specific AB for a specific AG
- Diversity: The immune system is capable of responding to a great variety of AGs
- Memory: Using memory B-cells, the immune system is capable of fighting repeated infections
- Clonal expansion: Increase the number of lymphocytes with high affinity of certain AGs
- Homeostasis: The immune system recover from an infection by itself
- No self-reactivity: The immune system does not attack the host body in the present of AGs

2.2. Germinal Center

When the body does not have a specific AB for an infection, the body starts the process of affinity maturation with the germinal center reaction. Germinal centers are micro-anatomical regions in the secondary lymph nodes, that form in the present of antigen [10]. The AGs that survive the other immunity mechanism arrives at the secondary lymph nodes, where are capture for the FDCs. The FDCs

activate near B-cells. The active B-cells end up being enclosed by the inactive ones forming a natural barrier that allow the active cells to proliferate, mutate and be selected.

The B-cells start to proliferate inside the GC, and compete for the antigen, this competition polarizes the GC in two distinct zones, the dark zone and the light zone. The dark zone is where B-cells proliferate through clonal expansion and somatic hyper-mutation, this process ensures the diversity of ABs. On the other hand, the light zone is where the B-cells are selected in accordance with their affinity.

On the light zone, the B-cells must find AG and internalize it, with the final purpose of digest the AG and expose their peptides to the Th-cell. The Th-cell gives a life signal allowing B-cells high affinity to live more time and therefore, proliferate and mutate with higher success.

In Figure 1, we show a schematic summary of the process. The GC reaction ensures the diversity through clonal expansion and somatic hyper-mutation in the dark zone, while in the light zone is a competitive process that reward the more adapted B-cells. The B-cells reentry to dark zone making this process an iterative refinement of the affinity [19,20].

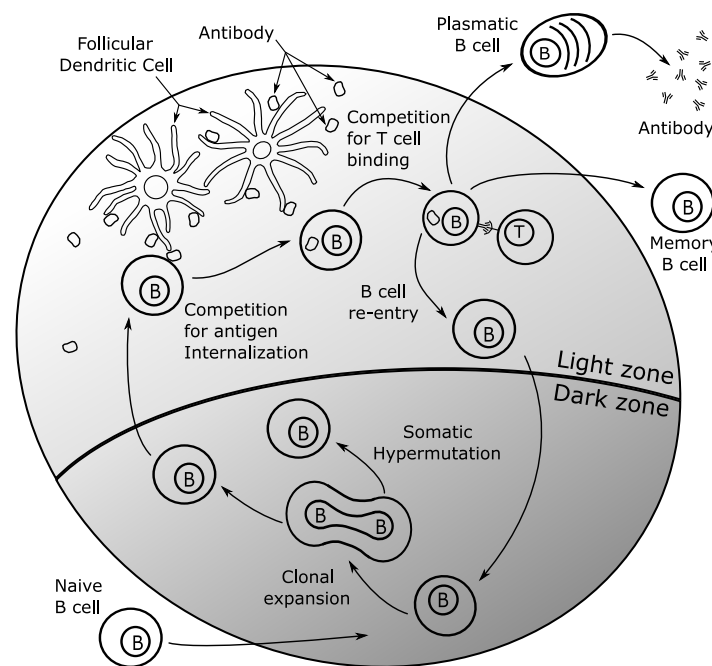


Figure 1. Germinal Center reaction.

Finally, when the GC generates high affinity B-cells for certain AG, some B-cells differentiate into plasmatic cells and release their ABs. A few B-cells become Memory B-cells that could live a long time and keeps information about this particular AG. Then, the GCs are capable of generating specific AB for a specific AG, and keep this information in Memory B-cells for future infections [21].

2.3. Algorithm Description

In this section, we explain the GCO algorithm. In Table 1, we present the computational analogy between germinal center and the optimization problem.

Table 1. Computational analogy between Germinal Center (GC) and Optimization.

Germinal Center	Optimization Problem
Antigen	Objective function
Antibody and B-cell	Candidate solution
Affinity	Objective function evaluation
T-help cell binding	Incrementation of life-signal

The GC reaction has multiple competitive processes, the GCO algorithm does not try to simulate GC reaction per se but to use some of its competitive mechanisms. A key factor in the GC reaction is the distinction between the dark zone and the light zone. The dark zone represents the diversification of the solutions that could be understood like a mutation process, many algorithms, such as Differential evolution and Genetic algorithms [1], already have the idea of mutation, but the dark zone includes not only a mutation process (somatic hyper-mutation), but also the clonal expansion. This clonal expansion is guided by the life signal of the B-cell, denoted by $\mathbb{L} \in [0, 100]$. The B-cells with greater life signal are more likely to clone, increasing the B-cell multiplicity.

In the light zone, B-cells with the best affinity are rewarded and the other cells age (lower their life signal). Then, the affinity-based selection in the light zone changes the probability of clone or death of a B-cell. In Figure 2, we present the GCO algorithm flowchart, where “For each B-cell” indicates that the process is applied in every candidate solution.

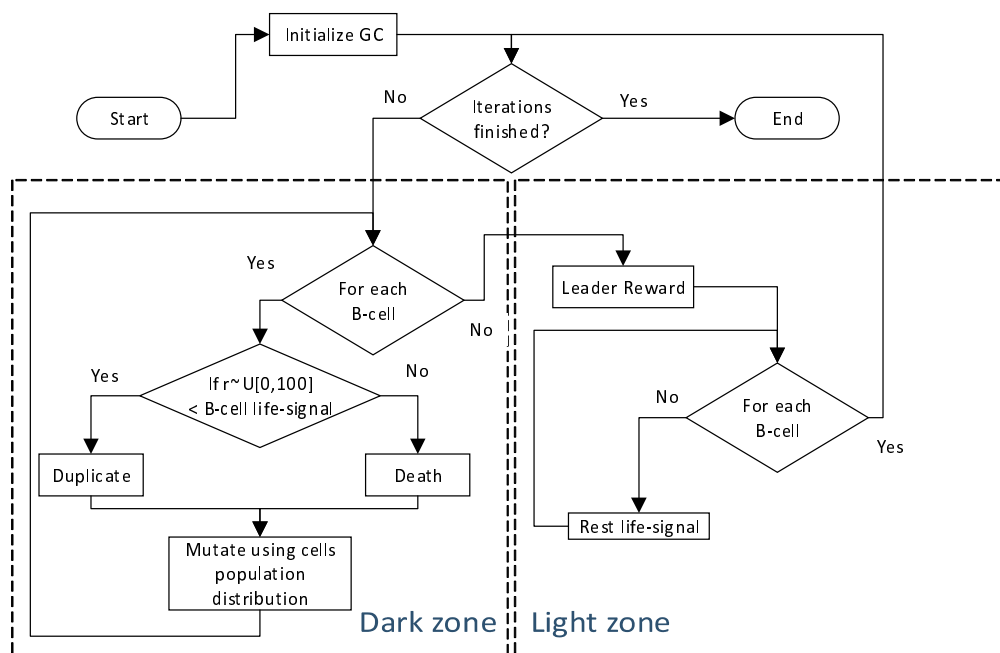


Figure 2. Germinal Center Optimization (GCO) algorithm flowchart.

As we are dealing with a population-based algorithm [1], there is a particular interest in the cells mutation and which information we use for crossover particles. In the GCO algorithm, we use the distribution of the cells multiplicity, denoted with \mathcal{C} , to select three individuals for crossover. It is important to note that initially, all the cells have a multiplicity of one, then the individuals are uniformly selected, but this distribution changes through iterations modeled by the competitive process. This kind of distribution offers different types of leadership behaviors in the collective intelligence, for example, initially the GCO algorithm behaves like Differential Evolution in the DE/rand/1 strategy [1], and when a particle wins for many iterations, the algorithm behaves like Particle Swarm optimization [1]. However, the leadership in GCO is not only dynamic, but it also includes temporal leadership, this is implemented when a particle mutates to a better solution, this new candidate substitutes the actual particle resetting the cell multiplicity to one.

Then, GCO algorithm offers a bio-inspired technique of adaptive leadership in collective intelligence algorithms for multivariate optimization problems. In Algorithm 1, we include an explicit pseudocode, where B_i is the i -esim B-cell, M is a mutant B-cell and a new candidate solution, $F \in [0, 2]$ is the mutation factor and $CR \in [0, 1]$ is the cross-ratio, \mathcal{C} is the distribution of cells multiplicity, and \mathbb{L} is the life signal.

Algorithm 1: GCO algorithm

```

Initialize B-cells ( $B_i$ )
foreach  $k \in \{1, \dots, Iterations\}$  do
    /* Dark-zone process */
    foreach  $i \in \{1, \dots, N\}$  do
        if  $r_i \sim U[0, 100] < \mathbb{L}$  of  $B_i$  then
            /* B-cell Duplication */
            Add one to GC cells counter
            Add one to  $B_i$  cells counter
        else
            /* B-cell Death */
            if Cells in  $B_i > 1$  then
                Rest one to GC cells counter
                Rest one to  $B_i$  cells counter
            end
        end
    end
    /* B-cell mutation */
    Calculate distribution  $\mathcal{C}$ 
    Using  $r_1, r_2, r_3 \sim \mathcal{C}$  choose 3 different B-cells:  $B_{r_1}, B_{r_2}$  and  $B_{r_3}$ 
    Create new Mutant  $M$ 
    foreach  $j \in \{1, \dots, D\}$  do
        if  $r \sim U[0, 1] < CR$  then
             $M(j) \leftarrow B_{r_1}(j) + F(B_{r_2}(j) - B_{r_3}(j))$ 
        else
             $M(j) \leftarrow B_i(j)$ 
        end
    end
    Evaluate objective function for  $M$ 
    if  $M$  is better than  $B_i$  then
        Substitute  $B_i$  for  $M$ 
        if  $M$  is the best then
            Save  $i$  index
        end
    end
    /* Light-zone process */
    Add 10 units to  $\mathbb{L}$  of the Best B-cell
    foreach  $i \in \{1, \dots, N\}$  do
        Rest 10 units to  $\mathbb{L}$  of  $B_i$ 
    end
end

```

3. Neural Inverse Optimal Control

In this section, we introduced the neural identifier based on a RHONN and its EKF training, and the inverse optimal control law designed. Consider the following affine discrete-time nonlinear system

$$\mathbf{x}(k) = f(\mathbf{x}(k)) + g(\mathbf{x}(k))\mathbf{u}(k) \quad (1)$$

where $\mathbf{x} \in \mathbb{R}^n$ is the state vector of the system, $\mathbf{u} \in \mathbb{R}^m$ is the control input vector, $f \in \mathbb{R}^n \rightarrow \mathbb{R}^n$ and $g \in \mathbb{R}^n \rightarrow \mathbb{R}^{n \times m}$ are smooth maps.

3.1. Neural Identification with Recurrent High Order Neural Networks (RHONN)

To identify the system (1), we used the following RHONN identifier based on a RHONN in series-parallel model:

$$\begin{aligned} \hat{\chi}_i(k+1) &= \omega_i^\top(k) \mathbf{z}_i(\mathbf{x}(k-l), \mathbf{u}(k)) \\ i &= 1, 2, \dots, n \end{aligned} \tag{2}$$

with

$$\mathbf{z}_i(\mathbf{x}(k), \mathbf{u}(k)) = \begin{bmatrix} z_{i_1} \\ z_{i_2} \\ \vdots \\ z_{i_{L_i}} \end{bmatrix} = \begin{bmatrix} \prod_{j \in I_1} \zeta_{i_j}^{d_{ij}(1)} \\ \prod_{j \in I_2} \zeta_{i_j}^{d_{ij}(2)} \\ \vdots \\ \prod_{j \in I_{L_i}} \zeta_{i_j}^{d_{ij}(L_i)} \end{bmatrix}, \quad \zeta_i = \begin{bmatrix} \zeta_{i_1} \\ \vdots \\ \zeta_{i_n} \\ \zeta_{i_{n+1}} \\ \vdots \\ \zeta_{i_{n+m}} \end{bmatrix} = \begin{bmatrix} S(x_1(k-l)) \\ \vdots \\ S(x_n(k-l)) \\ u_1(k) \\ \vdots \\ u_m(k) \end{bmatrix} \tag{3}$$

where $S(v) = 1/(1 + \exp(-\beta v))$, $\beta > 0$, n is the state dimension, $\hat{\chi}$ is state vector of the neural network, ω is the weight vector \mathbf{x} is the plant state vector, and $\mathbf{u} = [u_1, u_2, \dots, u_m]^\top$ is the input vector to the neural network.

The neural identifier (2) is presented in [14]. This neural identifier does not need previous knowledge of the model of the system, also, it does not need information of the disturbances and delays. Moreover, this model is semi-globally uniformly ultimately bounded (SGUUB) and the proof can be found in [14].

3.1.1. Training of RHONN with Extended Kalman Filter

The extended kalman filter estimates the state of a system with additive white noise in the input and in the output using a recursive solution in which each update of the state is estimated from the previous estimated state and the new input data [11,22].

For the case of neural networks the extended kalman filter training goal is to find the optimal weight vector which minimizes the prediction error. Due to the fact that the neural network mapping is non-linear the extended kalman filter (EKF) is required. The EKF-based training algorithm [11] is (4):

$$\omega_i(k+1) = \omega_i(k) + \eta_i \mathbf{K}_i(k) e_i(k) \tag{4}$$

$$\mathbf{K}_i(k) = \mathbf{P}_i(k) \mathbf{H}_i(k) [R_i(k) + \mathbf{H}_i^\top(k) \mathbf{P}_i(k) \mathbf{H}_i(k)]^{-1} \tag{5}$$

$$\mathbf{P}_i(k+1) = \mathbf{P}_i(k) - \mathbf{K}_i(k) \mathbf{H}_i^\top(k) \mathbf{P}_i(k) + \mathbf{Q}_i(k) \tag{6}$$

with

$$e_i(k) = x_i(k) - \hat{\chi}_i(k) \tag{7}$$

$$H_{ij} = \left[\frac{\partial \hat{\chi}_i(k)}{\partial \omega_{ij}(k)} \right]^\top \tag{8}$$

where $i = 1 \dots n$, $\omega_i \in \mathfrak{R}^{L_i}$ is the on-line adapted weight vector, $\mathbf{K}_i \in \mathfrak{R}^{L_i}$ is the Kalman gain vector, $e_i \in \mathfrak{R}$ is the identification error, $\mathbf{P}_i \in \mathfrak{R}^{L_i \times L_i}$ is the weight estimation error covariance matrix, χ_i is the i -th state variable of the neural network, $\mathbf{Q}_i \in \mathfrak{R}^{L_i \times L_i}$ is the estimation noise covariance matrix, $R_i \in \mathfrak{R}$ is the error noise covariance matrix and $\mathbf{H}_i \in \mathfrak{R}^{L_i}$ is a vector in which each entry H_{ij} is the derivative of the neural network state ($\hat{\chi}_i$) with respect to one neural network weight (ω_{ij}) and it is given by (8). \mathbf{P}_i and \mathbf{Q}_i are initialized as diagonal matrices with entries $\mathbf{P}_i(0)$ and $\mathbf{Q}_i(0)$, respectively. It is important to remark that $\mathbf{H}_i(k)$, $\mathbf{K}_i(k)$ and $\mathbf{P}_i(k)$ for the EKF are bounded.

3.1.2. Inverse Optimal Control

Optimal control finds a control law for a system such that a performance criterion is minimized. The criterion is a cost functional based on the state and control variables. The solution of the optimal control leads to the HJB equation which solution is not an easy task. Inverse optimal control is an alternative to optimal control, avoiding the HJB equation solution. For the inverse optimal control 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 cost functional, then, the CLF is modified in order to achieve asymptotic tracking for given trajectory references [15]. The existence of a CLF implies stability and every CLF can be considered as a cost functional. The CLF approach for control synthesis has been applied successfully to systems for which a CLF can be established, such as feedback linearizable, strict feedback and feed-forward ones [15].

The system (1) is supposed to have an equilibrium point $\mathbf{x}(0) = \mathbf{0}$. Moreover, the full state $\mathbf{x}(k)$ is assumed to be available. In order to ensure stability of the system (1) the following control Lyapunov function is proposed:

$$V(\mathbf{x}(k)) = \frac{1}{2}\mathbf{x}(k)^\top \mathbf{P}\mathbf{x}(k), \quad \mathbf{P} = \mathbf{P}^\top > 0 \quad (9)$$

The inverse optimal control law for the system (1) with (9) is:

$$\begin{aligned} \mathbf{u}(k) &= -\frac{1}{2}\mathbf{R}^{-1}(\mathbf{x}(k))g^\top(\mathbf{x}(k))\frac{\partial V(\mathbf{x}(k))}{\partial \mathbf{x}(k+1)} \\ &= -\frac{1}{2}(\mathbf{R}(\mathbf{x}(k)) + \frac{1}{2}g^\top(\mathbf{x}(k))\mathbf{P}g(\mathbf{x}(k)))^{-1}g^\top(\mathbf{x}(k))\mathbf{P}f(\mathbf{x}(k)) \end{aligned} \quad (10)$$

where $\mathbf{R}(\mathbf{x}(k)) = \mathbf{R}(\mathbf{x}(k))^\top > \mathbf{0}$ is a matrix whose elements can be functions of the system state or can be fixed. \mathbf{P} is a matrix such that the inequality (11) holds.

$$V_f(\mathbf{x}(k)) - \frac{1}{4}\mathbf{P}_1^\top(\mathbf{x}(k))(\mathbf{R}\mathbf{P}(\mathbf{x}(k)))^{-1}\mathbf{P}_1(\mathbf{x}(k)) \leq -\mathbf{x}^\top(k)\mathbf{Q}\mathbf{x}(k) \quad (11)$$

with

$$\mathbf{R}\mathbf{P}(\mathbf{x}(k)) = \mathbf{R}(\mathbf{x}(k)) + \mathbf{P}_2(\mathbf{x}(k)) \quad (12)$$

$$V_f(\mathbf{x}(k)) = \frac{1}{2}f^\top(\mathbf{x}(k))\mathbf{P}f(\mathbf{x}(k)) - V(\mathbf{x}(k)) \quad (13)$$

$$\mathbf{P}_1(\mathbf{x}(k)) = g^\top(\mathbf{x}(k))\mathbf{P}f(\mathbf{x}(k)) \quad (14)$$

$$\mathbf{P}_2(\mathbf{x}(k)) = \frac{1}{2}g^\top(\mathbf{x}(k))\mathbf{P}g(\mathbf{x}(k)) \quad (15)$$

$$\mathbf{Q} = \mathbf{Q}^\top > 0 \quad (16)$$

In [15], it is demonstrated that control law (10) is globally asymptotically stable. Moreover, (10) is inverse optimal in the sense that minimizes a cost functional [15].

4. Results

In this section, we briefly describe how GCO is applied to improve NIOC performance, we also show simulation results and real-time experimental results. The all-terrain tracked robot is a modified HD2[®] (HD2 is a registered trademark of SuperDroid Robots), shown in Figure 3. The changes of the modified HD2[®] are (Figure 4): the replacement of the original board for a system based on Arduino[®] (Arduino is a registered trademark of Arduino LLC), and an attachment of a wireless router. Chassis, tracks, batteries, and motors remained without modifications.



Figure 3. Modified HD2[®] Treaded All-Terrain Tracked Robot (ATR) Tank Robot Platform with wireless communication.

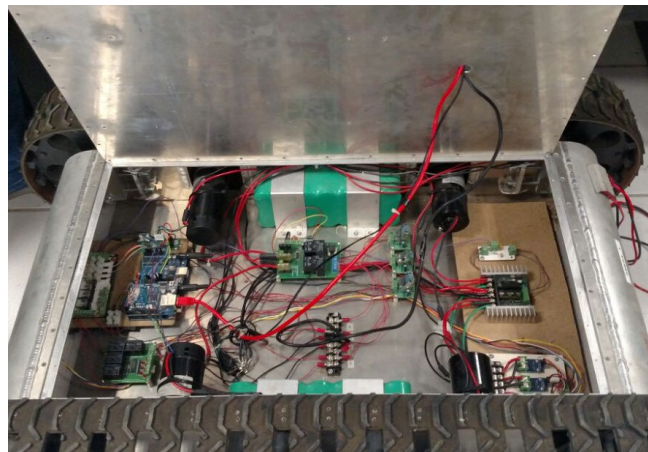


Figure 4. The interior of the Modified HD2[®] Treaded ATR Tank Robot.

4.1. Application to All-Terrain Tracked Robot Control

Considered as the most important type of mobile robots, a tracked robot runs on continuous tracks instead of wheels which develop a thrust higher than a wheeled robot. This kind of robots is ideal for working in tasks under rough terrains. Among the applications tracked robots can achieve are urban reconnaissance, forestry, mining, agriculture, rescue mission scenarios, autonomous planetary explorations [23–25].

A tracked robot consists of the following state variables [17,26,27] position x , position y , position θ , velocity 1, velocity 2, current 1 and current 2. In this work, we focus on the controller tracking performance for x , y and θ (Figure 5) for given references x_r , y_r and θ_r . The objective is to improve the NIOC results presented in [17] by using GCO to find the optimal parameters of the controller. These parameters are included in the matrices \mathbf{P}_1 and \mathbf{P}_2 defined in (11) and (12) respectively. The \mathbf{P}_1 and \mathbf{P}_2 are symmetric positive definite matrices, therefore we can define the set of variables $\{\psi_1, \psi_2, \psi_3, \psi_4, \psi_5\}$ for the optimization problem with the definition in (17).

$$\mathbf{P}_1 = \begin{bmatrix} \psi_1 & \psi_2 & \psi_3 \\ \psi_2 & \psi_1 & \psi_4 \\ \psi_3 & \psi_4 & \psi_1 \end{bmatrix} \quad \mathbf{P}_2 = \begin{bmatrix} \psi_5 & 0 \\ 0 & \psi_5 \end{bmatrix} \quad (17)$$

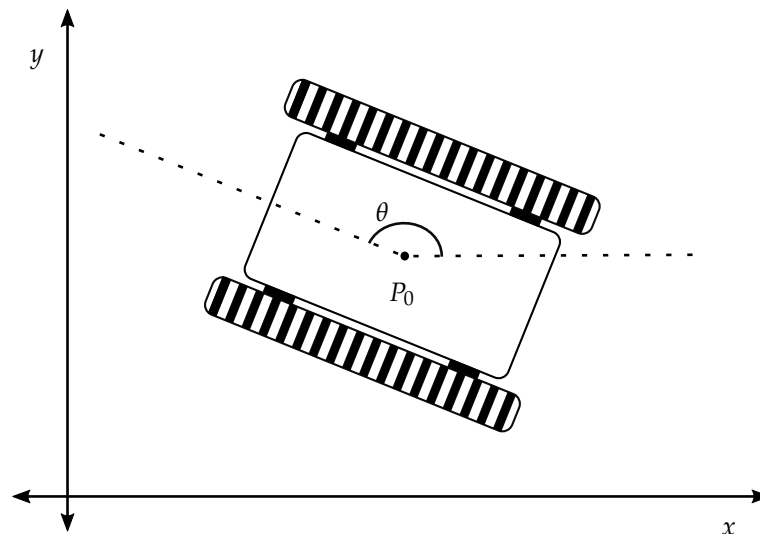


Figure 5. Schematic representation of a tracked robot, where x and y are the coordinates of P_0 and θ is the heading angle.

The lower bound of the search space is given by $\{1, 1, 1, 1, 1\}$ and the upper bound is given by $\{4 \times 10^7, 4 \times 10^7, 4 \times 10^7, 4 \times 10^7, 4 \times 10^4\}$. Next, the objective of the optimization is to get a better control tracking, in order to achieve this, we minimize the sum of the Root Mean Square Error (RMSE) in every state. This idea is described by (18), where n is the number of samples of the reference and estimated functions, we are using $n = 3335$ for real-time experiments and $n = 5000$ for simulation experiments.

$$f = \min_{\psi_1, \psi_2, \psi_3, \psi_4, \psi_5} \sqrt{\frac{\sum_{i=1}^n (\hat{x} - x)^2}{n}} + \sqrt{\frac{\sum_{i=1}^n (\hat{y} - y)^2}{n}} + \sqrt{\frac{\sum_{i=1}^n (\hat{\theta} - \theta)^2}{n}} \tag{18}$$

Then, the GCO algorithm will find optimal values that minimize (18). For the following experiments we use a GCO algorithm in five dimension running 150 iterations using 30 B-cells (4500 executions) for a test of 10 seconds; we set the parameters $F = 1.25$ and $CR = 0.7$. We include graphics for one simulation test using the all-terrain robot model presented in [17], and the results of one experimental test using the modified HD2[®] Treaded ATR Tank Robot Platform with wireless communication (Figure 3) presented in [17].

4.2. Simulation Results

In [17], \mathbf{P}_1 and \mathbf{P}_2 are defined as shown in (19) for simulation. We show in (20) \mathbf{P}_1^* and \mathbf{P}_2^* , which contain the parameters found by the GCO algorithm.

$$\mathbf{P}_1 = 14400 \begin{bmatrix} 162 & 1 & 2 \\ 1 & 162 & 3 \\ 2 & 3 & 162 \end{bmatrix} \quad \mathbf{P}_2 = 20 \begin{bmatrix} 20 & 0 \\ 0 & 20 \end{bmatrix} \tag{19}$$

$$\mathbf{P}_1^* = 1 \times 10^6 \begin{bmatrix} 5.7564 & 0 & 0 \\ 0 & 5.7564 & 0 \\ 0 & 0 & 5.7564 \end{bmatrix} \quad \mathbf{P}_2^* = \begin{bmatrix} 40000 & 0 \\ 0 & 40000 \end{bmatrix} \tag{20}$$

Figures 6–8 show the tracking performance and the error comparison for the position x , position y and position θ of the simulation test. For each figure, the graph on the left side shows the obtained signals for reference, the identified signal using NIOC [17], and the identified signal using the optimized

NIOC for its respective state variable. The graph on the right side shows the obtained the error signals for the NIOC [17] and the optimized NIOC for its respective state variable. This arrangement is maintained for all the following figures in this section.

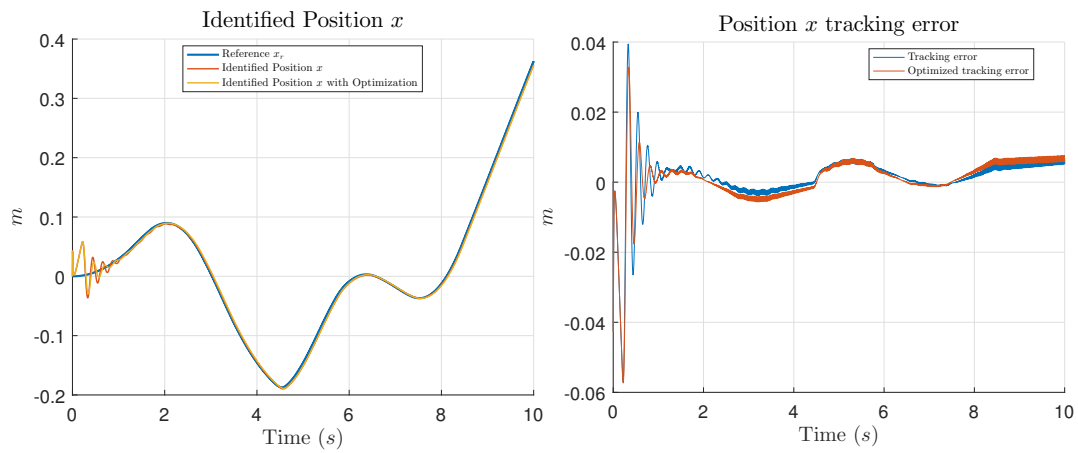


Figure 6. Tracking of x position (left) and error comparison (right) for the simulation test.

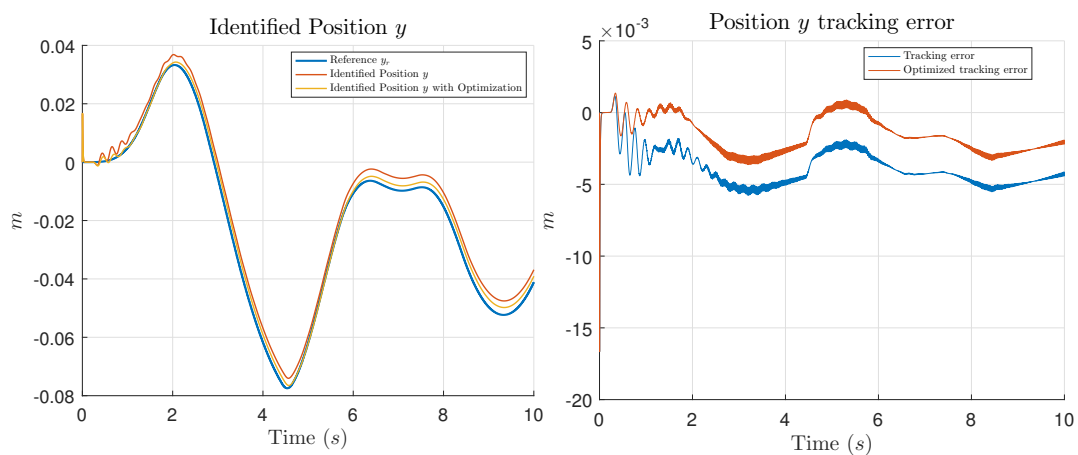


Figure 7. Tracking of y position (left) and error comparison (right) for the simulation test.

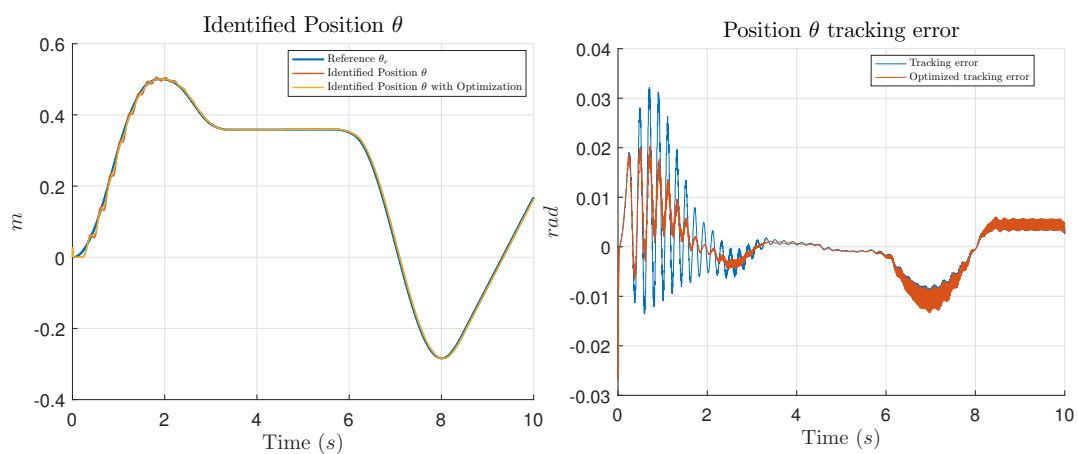


Figure 8. Tracking of θ (left) and error comparison (right) for the simulation test.

Table 2 shows the RMSE of each state variable and their total which is the evaluation of Equation (18) for this simulation test. Total does not have a physical meaning, it is the minimum found in the objective function in (18).

Table 2. Root Mean Squared Error (RMSE) in states for second simulation. Bold values highlight the best result.

Control	RMSE x (m)	RMSE y (m)	RMSE θ (rad)	Total
NIOC [17]	0.0073	0.0043	0.0064	0.0180
NIOC with GCO	0.0073	0.0025	0.0049	0.0147

A second simulation test was made resulting in the tracking errors shown in Table 3.

Table 3. Root Mean Squared Error (RMSE) in states for first simulation. Bold values highlight the best result.

Control	RMSE x (m)	RMSE y (m)	RMSE θ (rad)	Total
NIOC [17]	0.0077	0.0055	0.0064	0.0195
NIOC with GCO	0.0079	0.0041	0.0049	0.0169

Additionally to the presented results, in [17] it was demonstrated via simulations that the NIOC has a better performance than a super twisting scheme for a tracked robot model. The tracking comparison results shown in Table 4 are reported in [17].

Table 4. Tracking error comparison of NIOC [17] and a Super Twisting controller. Bold values highlight the best result.

RMSE	x (m)	y (m)	θ (rad)
NIOC [17]	0.0260	0.0362	0.0158
Super Twisting	0.0317	0.0036	0.0652

4.3. Real-Time Results

The work [17] presents a NOIC for the Modified HD2[®] Treaded ATR Tank Robot Platform with wireless communication (Figure 3), this implementation uses an RHONN identifier as (2) to identify the unknown model of the HD2[®]. The obtained model is then used as the based to synthesize the control law using the inverse optimal control approach. In [17] the values of \mathbf{P}_1 and \mathbf{P}_2 are defined in (21) for real-time operation. The following results show a comparison obtained with the values from [17] and \mathbf{P}_1^* and \mathbf{P}_2^* in (22) found by the GCO algorithm. In this section, there are presented two experiments named as “test 1” and “test 2”, respectively.

$$\mathbf{P}_1 = 72000 \begin{bmatrix} 162 & 1 & 2 \\ 1 & 162 & 3 \\ 2 & 3 & 162 \end{bmatrix} \quad \mathbf{P}_2 = 10000 \begin{bmatrix} 20 & 0 \\ 0 & 20 \end{bmatrix} \tag{21}$$

$$\mathbf{P}_1^* = 1 \times 10^7 \begin{bmatrix} 3.2039 & 0 & 4 \\ 0 & 3.2039 & 0 \\ 4 & 0 & 3.2039 \end{bmatrix} \quad \mathbf{P}_2^* = 1 \times 10^4 \begin{bmatrix} 2.4128 & 0 \\ 0 & 2.4128 \end{bmatrix} \tag{22}$$

Figures 9–11 show the tracking performance and the error comparison for the position x , position y and position θ of the experimental test 1.

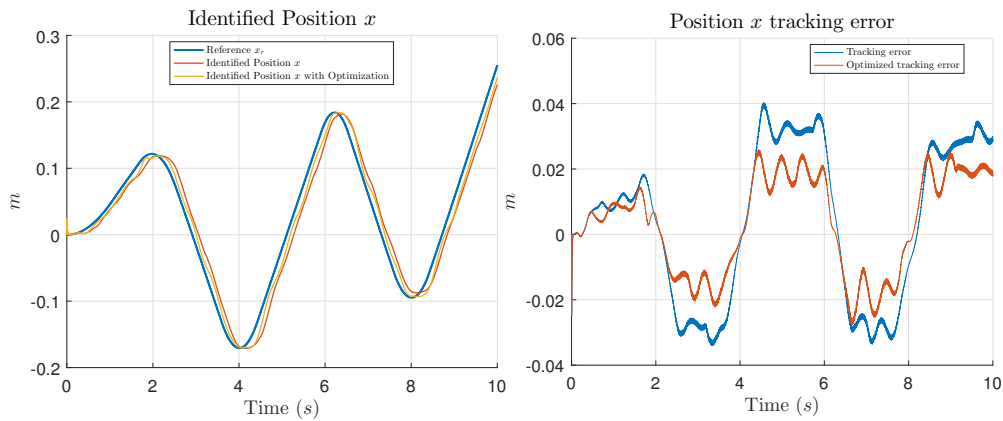


Figure 9. Tracking of x position (left) and error comparison (right) for the experimental test 1.

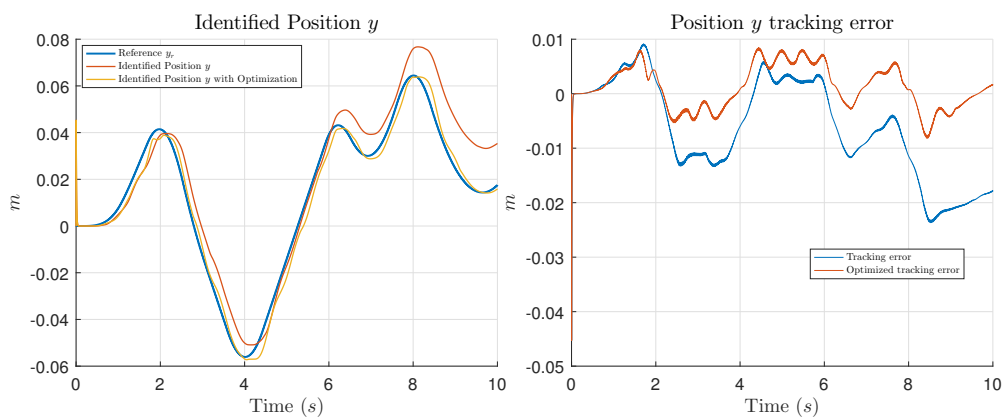


Figure 10. Tracking of y position (left) and error comparison (right) for the experimental test 1.

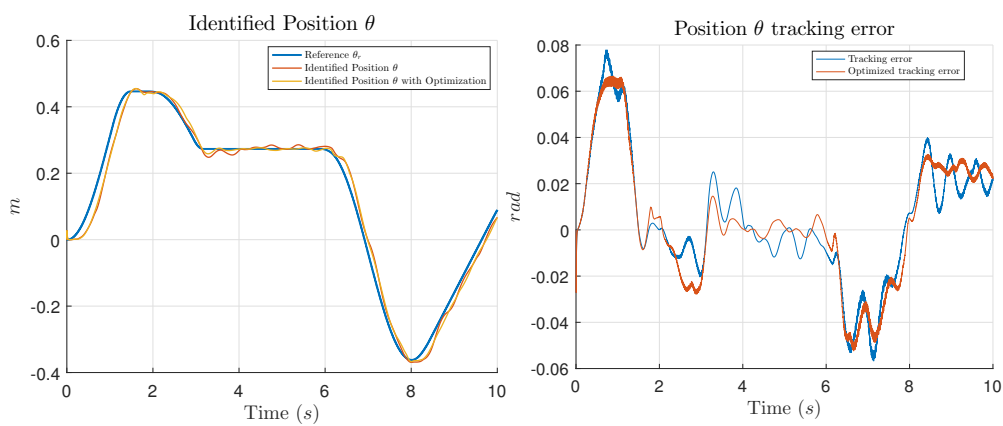


Figure 11. Tracking of θ (left) and error comparison (right) for the experimental test 1.

Table 5 shows the RMSE of each state variable and their total which is the evaluation of Equation (18) for the experimental test 1.

Table 5. Root Mean Squared Error in states for Real-Time experimental test 1. Bold values highlight the best result.

Control	RMSE x (m)	RMSE y (m)	RMSE θ (rad)	Total
NIOC [17]	0.0238	0.0109	0.0267	0.0614
NIOC with GCO	0.0152	0.0043	0.0270	0.0465

Figures 12–14 show the tracking performance and the error comparison for the position x , position y and position θ of the experimental test 2.

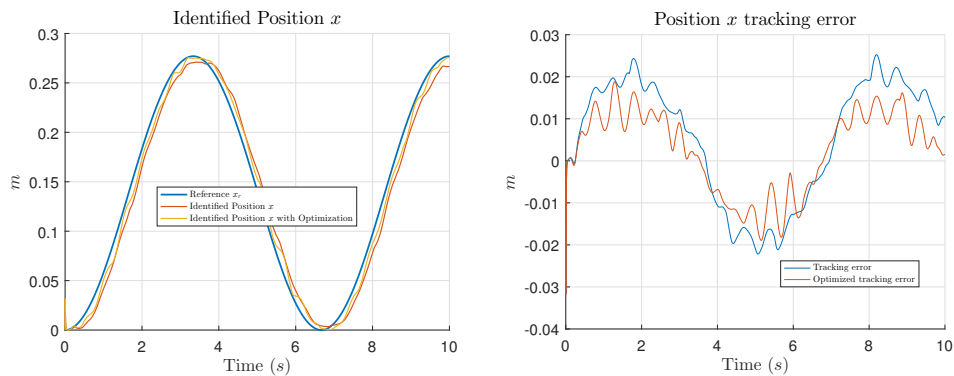


Figure 12. Tracking of x position (left) and error comparison (right) for the experimental test 2.

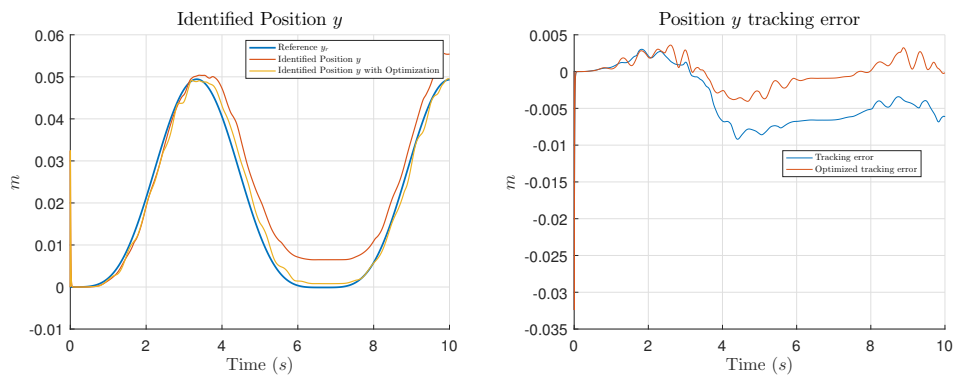


Figure 13. Tracking of y position (left) and error comparison (right) for the experimental test 2.

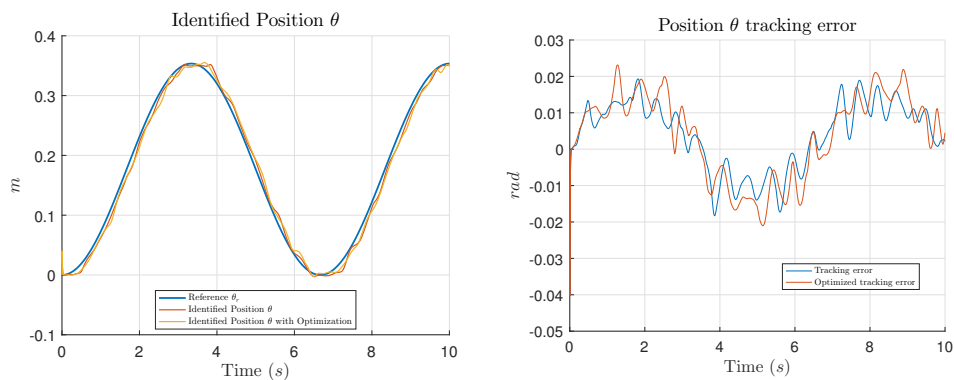


Figure 14. Tracking of θ (left) and error comparison (right) for the experimental test 2.

Table 6 shows the RMSE of each state variable and their total which is the evaluation of Equation (18) for the experimental test 2.

Table 6. Root Mean Squared Error (RMSE) in states for Real-Time experimental test 2. Bold values highlight the best result.

Control	RMSE x (m)	RMSE y (m)	RMSE θ (rad)	Total
NIOC [17]	0.0153	0.0052	0.0101	0.0305
NIOC with GCO	0.0100	0.0021	0.0123	0.0244

5. Conclusions

GCO is a hybridization between Evolutionary Computing and Artificial Immune System based on the Germinal Center reaction which is a biological process in vertebrates immune system that matures affinity of antibodies. GCO models a population of B-cells and the competitive process in their proliferation, then, a dynamic distribution of the cells multiplicity is constructed over the performance of the candidate solutions. This distribution allows to select B-cells for crossover with an adaptive leadership. The adaptive leadership takes the advantage of both high leadership and none leadership algorithms allowing it to find a better solution.

In this work, it is shown how GCO can help with the overall performance of a control technique like inverse optimal control, which depends on a number of designed parameters. Our results reveal a better performance of the controller version with the parameters obtained with GCO. It is also important to mention that the found parameters by GCO for the NIOC are not unique for a reference, those parameters can work for a number of references.

Acknowledgments: The authors thank the support of CONACYT Mexico, through Projects CB256769 and CB258068 (“Project supported by *Fondo Sectorial de Investigación para la Educación*”).

Author Contributions: Jorge D. Rios, Alma Y. Alanis and Carlos Lopez-Franco contribute to the NIOC design for an All-Terrain Tracked Robot. Carlos Villaseñor, Nancy Arana-Daniel and Esteban A. Hernandez-Vargas designed the GCO algorithm, the objective function and perform the optimization.

Conflicts of Interest: The authors declare no conflict of interest.

References

- Simon, D. *Evolutionary Optimization Algorithms*; John Wiley & Sons: Hoboken, NJ, USA, 2013.
- Haykin, S. *Neural Networks and Learning Machines*; Prentice Hall: Upper Saddle River, NJ, USA, 2009.
- Brabazon, A.; O’Neill, M.; McGarraghy, S. Artificial Immune Systems. In *Natural Computing Algorithms*; Springer: Berlin, Germany, 2015; pp. 301–332.
- Dasgupta, D.; Yu, S.; Nino, F. Recent advances in artificial immune systems: Models and applications. *Appl. Soft Comput.* **2011**, *11*, 1574–1587.
- Timmis, J.; Hone, A.; Stibor, T.; Clark, E. Theoretical advances in artificial immune systems. *Theor. Comput. Sci.* **2008**, *403*, 11–32.
- Abbas, A.K.; Lichtman, A.H.; Pillai, S. *Cellular and Molecular Immunology*; Elsevier Health Sciences: Amsterdam, The Netherlands, 2014.
- Ulutas, B.H.; Kulturel-Konak, S. A review of clonal selection algorithm and its applications. *Artif. Intell. Rev.* **2011**, *36*, 117–138.
- Dai, H.; Yang, Y.; Li, H.; Li, C. Bi-direction quantum crossover-based clonal selection algorithm and its applications. *Expert Syst. Appl.* **2014**, *41*, 7248–7258.
- Gao, S.; Dai, H.; Zhang, J.; Tang, Z. An expanded lateral interactive clonal selection algorithm and its application. *IEICE Trans. Fundam. Electron. Commun. Comput. Sci.* **2008**, *91*, 2223–2231.
- Victoria, G.D.; Nussenzweig, M.C. Germinal centers. *Ann. Rev. Immunol.* **2012**, *30*, 429–457.
- Sanchez, E.N.; Alanis, A.Y.; Loukianov, A.G. *Discrete-Time High Order Neural Control: Trained with Kalman Filtering*; Springer: Berlin, Germany, 2008.
- Alanis, A.; Sanchez, E. *Discrete-Time Neural Observers: Analysis and Applications*; Elsevier Science: Amsterdam, The Netherlands, 2017.

13. Zhang, M. *Artificial Higher Order Neural Networks for Economics and Business*; IGI Global Research Collection, Information Science Reference: Hershey, PA, USA, 2008.
14. Alanis, A.Y.; Rios, J.D.; Arana-Daniel, N.; Lopez-Franco, C. Neural identifier for unknown discrete-time nonlinear delayed systems. *Neural Comput. Appl.* **2016**, *27*, 2453–2464.
15. Sanchez, E.; Ornelas-Tellez, F. *Discrete-Time Inverse Optimal Control for Nonlinear Systems*; EBL-Schweitzer, CRC Press: Boca Raton, FL, USA, 2016.
16. Lopez, V.G.; Sanchez, E.N.; Alanis, A.Y.; Rios, J.D. Real-time neural inverse optimal control for a linear induction motor. *Int. J. Control* **2017**, *90*, 800–812.
17. Rios, J.D.; Alanis, A.Y.; Lopez-Franco, M.; Lopez-Franco, C.; Arana-Daniel, N. Real-time neural identification and inverse optimal control for a tracked robot. *Adv. Mech. Eng.* **2017**, *9*, 1687814017692970.
18. Di Noia, J.M.; Neuberger, M.S. Molecular mechanisms of antibody somatic hypermutation. *Annu. Rev. Biochem.* **2007**, *76*, 1–22.
19. Meyer-Hermann, M.; Mohr, E.; Pelletier, N.; Zhang, Y.; Vitorica, G.D.; Toellner, K.M. A theory of germinal center B cell selection, division, and exit. *Cell Rep.* **2012**, *2*, 162–174.
20. Zhang, Y.; Meyer-Hermann, M.; George, L.A.; Figge, M.T.; Khan, M.; Goodall, M.; Young, S.P.; Reynolds, A.; Falciani, F.; Waisman, A.; et al. Germinal center B cells govern their own fate via antibody feedback. *J. Exp. Med.* **2013**, *210*, 457–464.
21. Adachi, Y.; Onodera, T.; Yamada, Y.; Daio, R.; Tsuiji, M.; Inoue, T.; Kobayashi, K.; Kurosaki, T.; Ato, M.; Takahashi, Y. Distinct germinal center selection at local sites shapes memory B cell response to viral escape. *J. Exp. Med.* **2015**, *212*, 1709–1723.
22. Haykin, S. *Kalman Filtering and Neural Networks*; Adaptive and Cognitive Dynamic Systems: Signal Processing, Learning, Communications and Control; Wiley: New York, NY, USA, 2004.
23. Wong, J.; Huang, W. “Wheels vs. tracks”—A fundamental evaluation from the traction perspective. *J. Terramech.* **2006**, *43*, 27–42.
24. Siegwart, R.; Nourbakhsh, I.; Scaramuzza, D. *Introduction to Autonomous Mobile Robots*; Intelligent Robotics and Autonomous Agents; MIT Press: Cambridge, MA, USA, 2011.
25. González, R.; Rodríguez, F.; Guzmán, J. *Autonomous Tracked Robots in Planar Off-Road Conditions: Modelling, Localization, and Motion Control*; Springer: Berlin, Germany, 2014.
26. Das, T.; Kar, I.N. Design and implementation of an adaptive fuzzy logic-based controller for wheeled mobile robots. *IEEE Trans. Control Syst. Technol.* **2006**, *14*, 501–510.
27. Moosavian, S.A.A.; Kalantari, A. Experimental slip estimation for exact kinematics modeling and control of a Tracked Mobile Robot. In Proceedings of the 2008 IEEE/RSJ International Conference on Intelligent Robots and Systems, Nice, France, 22–26 September 2008; pp. 95–100.



© 2017 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (<http://creativecommons.org/licenses/by/4.0/>).