Applied Mathematical Sciences, Vol. 10, 2016, no. 13, 637 - 652
HIKARI Ltd, www.m-hikari.com
http://dx.doi.org/10.12988/ams.2016.510653

Planification of an Optimal Path for a Mobile Robot Using Neural Networks

M. Khouil

Distributed Signals and Systems, and Artificial Intelligence (SSDIA) Laboratory
ENSET, Hassan II University of Casablanca, Morocco

I. Sanou

Distributed Signals and Systems, and Artificial Intelligence (SSDIA) Laboratory
ENSET, Hassan II University of Casablanca, Morocco

M. Mestari

Distributed Signals and Systems, and Artificial Intelligence (SSDIA) Laboratory
ENSET, Hassan II University of Casablanca, Morocco

A. Aitelmahjoub

ENSAM Casablanca
Avenue Nile 150 Casablanca, Morocco

Abstract

The objective of the present work is to find an optimal path for a mobile of four wheels. The kinematic model related to the studied engine is a nonlinear system, where several nonlinear objective functions must be optimized in a conflicting situation. Under these circumstances, we propose to apply the algorithm studied in [13] for the planification of this optimal trajectory. For more efficiency, we took advantage of artificial neural networks parallelism by using neurons commonly used seen in [14], [13] and some other new created ones. Also a Matlab

simulation has been programmed in the last section toward observing the convergence of the results obtained.

Keywords: Optimal path, Mobile Robot, Nonlinear equality constrained multiobjective optimization problem, Neural networks, Simulation

1 Introduction

In robotics field, path planning has long been a fundamental problem. Although many solutions have been proposed [11], [10], there is always room for more research to reduce the cost and time calculation. The problem is to find an optimal trajectory leading the robot from an initial configuration to an arrival one within the constraints and taking into account the geometrical uncertainties. The main cause limiting the development of robotic systems is the computational complexity [16]. Direct application of classic optimization methods to Nonlinear Equality Constrained Multiobjective Optimization Problems [13] is mostly difficult and complicated to solve. These methods often makes use of very complex mathematical tools, also the amount of computation required may grow exponentially with the problem size. A number of new approaches using genetic algorithms has been proposed in several manners to solve NECMOP (for example, see [1] - [7]). Unfortunately, when the constraints of the problem considered becomes too laborious to satisfy or when the objective space is non convex, the multiobjective genetic algorithms converge with arduousness to optimal pareto front [2] and [9]. Besides, the algorithm we are using in this study has brought some advances in terms of swiftness and simplicity. This approach makes use of decomposition coordination principle which allows nonlinearity to be treated at a local level and where coordination is achieved through use of Lagrange multipliers [15]. Artificial neural network are commonly used in the area of different classes of optimization problems. Analog neural networks has the faculty to process a large number of variables simultaneously, which makes possible to find solutions for complex multiobjective optimization problems in real time. In furtherance of simplifying the use of the algorithm studied in this paper and to subsequently put the mobile robot studied theoretically into practice in a real operational environment, we have developed a simulation comprising two examples of optimal trajectory using the Matlab software. The complete Matlab code has a cinch structure using basic functions. The scientific contribution of this study is the application of a new method of resolution of NECMOPs in order to conceive an optimal path. To reduce the time and complexity of computation, we used the artificial neural network inside an architecture based on simple circuit. The remainder of this paper will be as follows: The second section concerns the planification path of the robot studied, then in a third section the resolution

of the kinematic model seen in section one according to the decomposition-coordination method. The forth section is for the construction of the neural network Architecture and finally a fifth section for the Matlab simulation.

2 PLANNING PATH FOR A MOBILE ROBOT

2.1 Study of the environment and its constraints

The studied robot is considered as rigid and is evolving on a plane. It has conventional wheels: the contact point between the wheel and the ground is reduced to a point I and the wheel is subjected to the constraint of rolling without slipping. Under good conditions, there is rolling without sliding of the wheel on the ground, however, the relative velocity of the wheel in relation to ground at the point of contact is zero.
Let P be the wheel center, Q the contact point linking the wheel with the ground, ψ is the wheel's clean rotation angle and θ the angle between the wheel plane and the ground one$(O, \overrightarrow{x}, \overrightarrow{z})$ (See fig.1). The nullity of the relative speed $\overrightarrow{V}_{Q(wheel/ground)}$ at the contact point gives a vector relationship between the speed $\overrightarrow{V}_P$ of the wheel's center P and the vector speed of wheel's rotation $\overrightarrow{W}$:

$$\overrightarrow{V}_Q = \overrightarrow{V}_P + \overrightarrow{W} \wedge \overrightarrow{PQ} = \overrightarrow{0} \tag{1}$$

2.2 Kinematic model of a mobile robot of four wheels

Take as a reference point (x, y) the mid-axis wheels of the rear axle where both wheels matrices is located (see fig.2). We introduce here the notion of director wheel center. Its introduction can simplify the equations by disregarding the steering wheels coupling mechanism to observe the rolling without slipping constraints and considering only one steering angle. The corresponding kinematic model:

$$\begin{cases} \dot{x} = & V\cos\theta \\ \dot{y} = & V\sin\theta \\ \dot{\theta} = & \dfrac{V}{L}\tan\psi \\ \dot{\psi} = & W \end{cases} \tag{2}$$

With $(\dot{x}, \dot{y}, \dot{\theta}, \dot{\psi})$ the first derivative with respect to time. where L represents the distance between the front axes and rear wheels, Ψ is the steering angle which is formed by the front wheels and the vehicle main axis, V is the linear velocity, and W is the angular velocity according to the vertical axis of the steering wheel relative to the vehicle body.

2.3 Nonlinear formatting of the Kinematic model

Our aim is to compute an optimal path in a short time following a specialized algorithm solving nonlinear systems with multiple constraints. This algorithm will be used for the first time in robotics field.
In what follows we notice the transformation of the kinematic model (2) with respect to NECMOP system format [13]. We converted the system of differential equations (derivations with respect to time) into an appropriate system of difference equations using the forward Euler rule as follows:

$$\begin{cases} x_{k+1} = & V_k \cos\theta_k \delta t \\ y_{k+1} = & V_k \sin\theta_k \delta t \\ \theta_{k+1} = & \dfrac{V_k}{L} \tan\psi_k \delta t + \theta_k \\ \psi_{k+1} = & W_k \delta t + \psi_k \end{cases} \tag{3}$$

As a matter of simplification we put:

$$q_{k+1} = f(q_k, u_k) \tag{4}$$

with $q_k = \begin{pmatrix} x_k \\ y_k \\ \theta_k \\ \psi_k \end{pmatrix}$ *and* $U_k = \begin{pmatrix} V_k \\ W_k \end{pmatrix}$.

U_k represents the control function and q_k denotes the state of the system at time k.

3 Resolution of the nonlinear system

3.1 Analysis of the problem

With the objective of applying the algorithm that solves nonlinear systems with multiple constraints, we set the following system:

$$\begin{cases} min\, E(q,u) & \\ Z_k = f(q_k, u_k) & k = 0, 1, ..., N-2 \\ \quad and & \\ q_k = Z_{k-1} & k = 1, 2, ..., N-1 \\ \quad with & \\ q_0 = q(0) & \end{cases} \tag{5}$$

where :

- E(q,u) represent the energy function
- $q = [q_0^T, q_1^T, ..., q_N^T]^T$ and $u = [u_0^T, u_1^T, ..., u_{N-1}^T]^T$
- Z_k is the output of the system and $q(0)$ is the given initial condition

For the moment, we proceed with the construction of the ordinary Lagrange function:

$$L = \sum_{k=0}^{N-1} L_k = \sum_{k=0}^{N-1} \frac{1}{N} E(q,u) + \mu_k^T (f(q_k, u_k) - Z_k) + \beta_k^T (q_k - Z_{k-1}) \quad (6)$$

μ_k and β_k are composed of four components and represents the Lagrange multiplier Vectors, their main function is to manage the equality constraints in (5). we note that q_k and $f(q_k, u_k)$ are also composed of four components, where u_k is composed of two:

$$q_k^T = (x_k, y_k, \theta_k, \psi_k),\ u_k^T = (V_k, W_k) \quad (7)$$

and

$$f(q_k, u_k) = (f_1(q_k, u_k), f_2(q_k, u_k), f_3(q_k, u_k), f_4(q_k, u_k)) \quad (8)$$

The derivations of the ordinary Lagrange function enable us to transform the equality constrained minimization problem (5) into a set of differential equations. An equilibrium point $(q_k^*, u_k^*, \mu_k^*, \beta_k^*, Z_k^*)$, with respect to the KKT conditions [12], satisfies the following equations:

$$\nabla_{q_k} L = \frac{1}{N} \frac{\delta E}{\delta q_k} + \frac{\delta f^T}{\delta q_k} \mu_k^* + \beta_k^* = 0 \quad For \quad 1 \le k \le N-1 \quad (9)$$

$$\nabla_{u_k} L = \frac{1}{N} \frac{\delta E}{\delta u_k} + \frac{\delta f^T}{\delta u_k} \mu_k^* = 0 \quad For \quad 0 \le k \le N-1 \quad (10)$$

$$\nabla_{z_k} L = -\mu_k^* - \beta_k^* = 0 \quad For \quad 0 \le k \le N-2 \quad (11)$$

$$\nabla_{\mu_k} L = f(q_k^*, u_k^*) - Z_k^* = 0 \quad For \quad 0 \le k \le N-1 \quad (12)$$

$$\nabla_{\beta_k} L = q_k^* - Z_{k-1}^* = 0 \quad For \quad 1 \le k \le N-1 \quad (13)$$

These five differential equations above (9)-(13) serves as the key of solving the equality constrained minimization problem (5).
In consonance with the kinematic model of the mobile robot studied here. We listed below all the derivations seen in the differential equations:

Derivation of the function f with respect to q_k

$$\frac{\delta f}{\delta q_k} = \left(\frac{\delta f_1}{\delta q_k}, \frac{\delta f_2}{\delta q_k}, \frac{\delta f_3}{\delta q_k}, \frac{\delta f_4}{\delta q_k}\right) \tag{14}$$

with:

$$\frac{\delta f_1(q_k^*, u_k^*)}{\delta q_k} = \begin{pmatrix} 1 \\ 0 \\ -V_k \sin\theta_k \delta t \\ 0 \end{pmatrix} \tag{15}$$

$$\frac{\delta f_2(q_k^*, u_k^*)}{\delta q_k} = \begin{pmatrix} 0 \\ 1 \\ V_k \cos\theta_k \delta t \\ 0 \end{pmatrix} \tag{16}$$

$$\frac{\delta f_3(q_k^*, u_k^*)}{\delta q_k} = \begin{pmatrix} 0 \\ 0 \\ 1 \\ \frac{V_k}{L}(1 + (\tan\psi_k)^2)\delta t \end{pmatrix} \tag{17}$$

$$\frac{\delta f_4(q_k^*, u_k^*)}{\delta q_k} = \begin{pmatrix} 0 \\ 0 \\ 0 \\ 1 \end{pmatrix} \tag{18}$$

Derivation of the function f with respect to u_k

$$\frac{\delta f}{\delta u_k} = \left(\frac{\delta f_1}{\delta u_k}, \frac{\delta f_2}{\delta u_k}, \frac{\delta f_3}{\delta u_k}, \frac{\delta f_4}{\delta u_k}\right) \tag{19}$$

with :

$$\frac{\delta f_1(q_k^*, u_k^*)}{\delta u_k} = \begin{pmatrix} \cos\theta_k \delta t \\ 0 \end{pmatrix} \tag{20}$$

$$\frac{\delta f_2(q_k^*, u_k^*)}{\delta u_k} = \begin{pmatrix} \sin\theta_k \delta t \\ 0 \end{pmatrix} \tag{21}$$

$$\frac{\delta f_3(q_k^*, u_k^*)}{\delta u_k} = \begin{pmatrix} \frac{\tan\psi_k}{L}\delta t \\ 0 \end{pmatrix} \tag{22}$$

$$\frac{\delta f_4(q_k^*, u_k^*)}{\delta u_k} = \begin{pmatrix} 0 \\ \delta t \end{pmatrix} \tag{23}$$

Regarding the energy function E, which is expressed by only the control function u_k, we notice its derivation with respect to q_k:

$$\frac{\delta E(q_k^*, u_k^*)}{\delta q_k} = \begin{pmatrix} 0 \\ 0 \\ 0 \\ 0 \end{pmatrix} \tag{24}$$

Derivation of the energy function E with respect to u_k:

$$\frac{\delta E(q_k^*, u_k^*)}{\delta u_k} = \frac{1}{2} \begin{pmatrix} V_k \\ W_k \end{pmatrix} \tag{25}$$

3.2 Decomposition-coordination solving method

The only efficient method for solving the equality constrained minimization problem (5) enclose the decomposition of the associated treatment system into differential equations (9)-(13) between two levels. The processing of the system (9)-(13) is partioned into two levels. The upper level is in charge of equations (11) and (13) to fix $Z_k (0 \leq k \leq N-2)$ and $\beta_k (1 \leq k \leq N-1)$ and then propose it to the lower level. However each subproblem becomes:

$$\begin{cases} Z_k (k = 1, 2, ..., N-2) \; and \; \beta_k (k = 1, 2, ..., N-2) \\ \quad Which \; is \; given \; by \; the \; upper \; level \\ minimize \; E(q, u) + \beta_k^T q_k - \beta_k^T Z_{k-1} \\ subject \; to \; Z_k = f(q_k, u_k) \end{cases} \tag{26}$$

Resultantly, the solution of each sub problem is equal to the processing of equations (9), (10) and (12) for $Z_k (k = 0, 1, ..., N-2)$ and $\beta_k (k = 1, ..., N-1)$ supplied by the upper level. Hence, with the application of a gradient method we obtain the system of difference equations using the Forward Euler rule:

$$q_{ks}^{(l+1)} = q_{ks}^{(l)} - \lambda_q \left(\frac{1}{N} \frac{\delta E^{(l)}}{\delta q_{ks}} + \sum_{i=1}^{4} \frac{\delta f_i^{(l)}}{\delta q_{ks}} \mu_{ki}^{(l)} + \beta_{ks}^{(j)} \right) \tag{27}$$
$$k = 1, ..., N-1 \qquad and \qquad s = 1, ..., 4$$

$$u_{ks}^{(l+1)} = u_{ks}^{(l)} - \lambda_u \left(\frac{1}{N} \frac{\delta E^{(l)}}{\delta u_{ks}} + \sum_{i=1}^{4} \frac{\delta f_i^{(l)}}{\delta u_{ks}} \mu_{ki}^{(l)} \right) \tag{28}$$
$$k = 1, ..., N-1 \qquad and \qquad s = 1, 2$$

$$\mu_{ks}^{(l+1)} = \mu_{ks}^{(l)} + \lambda_\mu \left(f_s(q_k^{(l)}, u_k^{(l)}) - Z_{ks}^{(j)} \right) \tag{29}$$
$$k = 0, ..., N-1 \qquad and \qquad s = 1, ..., 4$$

Where $\lambda_q, \lambda_u, \lambda_\mu$ are all stricly positive. This system of difference equations (27)-(29) can only be solved locally if the necessary information about $\beta_k^{(j)}(k = 1, ..., N-1)$ and $Z_k^{(j)}(k = 0, ..., N-2)$ sent from the upper level is attainable. The upper level works simultaneously on $\beta_k^{(j)}(k = 1, ..., N-1)$ and $Z_k^{(j)}(k = 0, ..., N-2)$, which make up the coordination parameters. These coordination are considered as known within the lower level, allowing local resolution of the system of difference equations (27)-(29) and determination of the variables $q_k^*(Z_k^{(j)}, \beta_k^{(j)})$, $u_k^*(Z_k^{(j)}, \beta_k^{(j)})$ and $\mu_k^*(Z_k^{(j)}, \beta_k^{(j)})$ which respectively satisfy equations (9), (10) and (12). The results $q_k^*(Z_k^{(j)}, \beta_k^{(j)})$ and $\mu_k^*(Z_k^{(j)}, \beta_k^{(j)})$ are supplied for the upper level which verifies if the previously supplied information was correct and corrects it if necessary. At this stage the lower level can recommence its work with information of increased validity. This correction is necessary for the evolution and satisfaction of equations (11) and (13).
The upper level proceeds progressively making the necessary correction to the coordination parameters in order to approach the satisfaction of coordination equations (9) and (11). Though, the coordination parameters at iteration j+1 are improvements of the coordination parameters at iteration j (see fig.3).

$$\begin{aligned} &Z_{ks}^{(j+1)} = Z_{ks}^{(j)} - \lambda_Z(-\mu_{is}^*(Z_k^{(j)}, \beta_k^{(j)}) - \beta_{k+1,s}^{(j)}) \\ &k = 0, 1, ..., N-2 \qquad and \qquad s = 1, 2, ..., 4. \end{aligned} \tag{30}$$

$$\begin{aligned} &\beta_{ks}^{(j+1)} = \beta_{ks}^{(j)} + \lambda_\beta(q_{ks}^*(Z_k^{(j)}, \beta_k^{(j)}) - Z_{k-1,s}^{(j)}) \\ &k = 1, 2, ..., N-1 \qquad and \qquad s = 1, ..., 4 \end{aligned} \tag{31}$$

with λ_Z and λ_β strictly positive.

The solution of the system (27)-(29) is repeated until we reach the coordination satisfactory ,i.e satisfaction of equations (11) and (13).

3.3 Analysis of the stability

Regarding the convergence, we have seen according to [13] that the two sufficient conditions for stability were:

1. Only one of the matrices $\frac{\delta G_k^{*T}}{\delta v_k}$ (k=0,1,...,N-1) must be absolutely positive definite against the other matrices can be only positive semi-definite. Where:
$$G_k = \begin{pmatrix} \nabla_{q_k} L \\ \nabla_{u_k} L \end{pmatrix} \quad and \quad v_k = \begin{pmatrix} q_k \\ u_k \end{pmatrix} \tag{32}$$
2. The adaptive coefficient λ must be chosen in a way that:
$$0 < \lambda < \left| \frac{B(j)}{A(j)} \right| \tag{33}$$

Where:

$$A(j) = \sum_{k=0}^{N-1} H_k^{(j)T} H_k^{(j)} + R_k^{(j)T} R_k^{(j)} \tag{34}$$

and

$$\begin{aligned} B(j) = \sum_{k=0}^{N-1} (-e_{z_k}^{(j)T} \frac{\delta H_k^*}{\delta \mu_k} e_{\mu_k}^{(j)} - e_{z_k}^{(j)T} \frac{\delta H_k^*}{\beta_{k+1}} e_{\beta_{k+1}}^{(j)} \frac{\delta R_k^*}{\delta v_k} e_{v_k}^{(j)} \\ + e_{\beta_k}^{(j)T} \frac{\delta R_k^*}{\delta z_{k-1}} e_{z_{k-1}^{(j)}}) \end{aligned} \tag{35}$$

with $H_k = \nabla_{z_k} L$ *and* $R_k = \nabla_{\beta_k} L$

and $e_{v_k}^{(j)}$, $e_{Z_k}^{(j)}$, $e_{\beta_k}^{(j)}$, $e_{\mu_k}^{(j)}$ designate the errors calculated at iteration j of the coordination loop. For this second condition, we took a variable adaptive coefficient λ that is adjusted at each iteration j of the coordination loop, with α a parameter which can take any value within the open interval $]1; 0[$ so that:

$$\lambda = \alpha \left| \frac{B(j)}{A(j)} \right| \tag{36}$$

The convergence speed of the algorithm depends essentially on how adaptive coefficient λ is chosen at each iteration of the coordination loop.

4 Modeling algorithm using Artificial Neural Networks

4.1 Neural networks used

The neural network here studied is composed of multiple networks such as weighted network (WN), the Jacobian network, the lower and upper level network and the Input network. Each network is himself composed of some specific and common neurons.

The WN network is specialized in computing the weighted sum of objective functions $E(q, u)$. This network is realized by incorporating adaptive nonlinear building blocks and a linear neuron. The linear neuron is commonly used in neural networks applications [14], [13], [15]. It's representation is shown in Fig.4, where y is the output, θ ($\theta \in \Re$) is the external threshold, w_i are the synaptic weights, x_i are the inputs ($i = 1, 2, ..., n$) and n is the number of inputs. The linear neuron sums the n weighted inputs and pass the result through a linear activation function according to the equation:

$$y = \phi_L \left(\sum_{i=1}^{n} w_i x_i - \theta \right) \tag{37}$$

where ϕ_L is the linear activation function, defined by $\phi_L(x) = x$.
The two Jacobian networks JNN developed in [15] are used for the final architecture (see fig.5) and are respectively related to the calculation of the jacobians J_F and J_E specific to the dynamic system and the energy function given in (5).
The Jacobians matrices J_F and J_E are defined respectively as the $(4+2) \times 4$ and $(4+2) \times 1$ matrices:

$$J_F = \begin{pmatrix} \frac{\delta f_1}{\delta x_k} & \frac{\delta f_1}{\delta y_k} & \frac{\delta f_1}{\delta \theta_k} & \frac{\delta f_1}{\delta \psi_k} & \frac{\delta f_1}{\delta V_k} & \frac{\delta f_1}{\delta W_k} \\ \frac{\delta f_2}{\delta x_k} & \frac{\delta f_2}{\delta y_k} & \frac{\delta f_2}{\delta \theta_k} & \frac{\delta f_2}{\delta \psi_k} & \frac{\delta f_2}{\delta V_k} & \frac{\delta f_2}{\delta W_k} \\ \frac{\delta f_3}{\delta x_k} & \frac{\delta f_3}{\delta y_k} & \frac{\delta f_3}{\delta \theta_k} & \frac{\delta f_3}{\delta \psi_k} & \frac{\delta f_3}{\delta V_k} & \frac{\delta f_3}{\delta W_k} \\ \frac{\delta f_4}{\delta x_k} & \frac{\delta f_4}{\delta y_k} & \frac{\delta f_4}{\delta \theta_k} & \frac{\delta f_4}{\delta \psi_k} & \frac{\delta f_4}{\delta V_k} & \frac{\delta f_4}{\delta W_k} \end{pmatrix} \tag{38}$$

and

$$J_E = \begin{pmatrix} \frac{\delta E}{\delta x_k} & \frac{\delta E}{\delta y_k} & \frac{\delta E}{\delta \theta_k} & \frac{\delta E}{\delta \psi_k} & \frac{\delta E}{\delta V_k} & \frac{\delta E}{\delta W_k} \end{pmatrix} \tag{39}$$

The input network is realized by using the switched capacitor techniques putting into practice some elementary and simple elements as seen in [13].
Concerning the Upper level and lower level networks, their architecture follow the exact reasoning and the same concatenation seen in Fig.3. The decomposition-coordination method illustrated in Fig.3 can be executed practically by using switched capacitor techniques [15]. The final architecture of the neural network is detailed in Fig.5.

5 Simulation of the decomposition-coordination algorithm by matlab

The simulation will be elaborated with the Matlab software. During this study we observed that the coordination parameter (λ) and the shape of the trajectory change constantly according to the initial condition values and number of steps chosen. Therefore, we took two different examples that gives us two different trajectories (see Fig.6, Fig.7) and two variations of λ (see Fig.8, Fig.9). Indeed, as explained before, the curves (Fig.8)-(Fig.9) show that under the conditions of stability, convergence is guaranteed. The method implemented for selecting adaptive coefficients λ has greatly improved the convergence speed and performs the algorithm stability calculation speed compromise with much information and not intuitive way.
The analysis therefore shows that if λ is chosen too small, the convergence

can be very slow, so it is advantageous to initially choose λ sufficiently large. Since initially the ratio $\left|\frac{B(0)}{A(0)}\right|$ is sufficiently high, the initial value chosen for λ is large enough.

6 Conclusion

The aim of our research in the present paper is about the theoretical implementation of the decomposition-coordination method for the case of a mobile robot with four wheels for the sake of planning an optimal path.
The conventional optimization techniques are established commonly on a global treatment inserted within an iterative process. The modification of the entire problem and its resolution are in most of the case an expensive stage in time and memory space of the treatment process, making it almost impossible to treat these problems in real time.
With regard to overcome the constraint of time and memory, we used the artificial neural networks in order to process in a parallel and appropriate time. The neural network architecture is based on simple circuit element according to VLSI techniques.
Finally we implemented the decomposition-coordination method in Matlab software by taking two effective examples with a view to test the evolution of the coordination parameter and to evaluate the optimal path resulted.

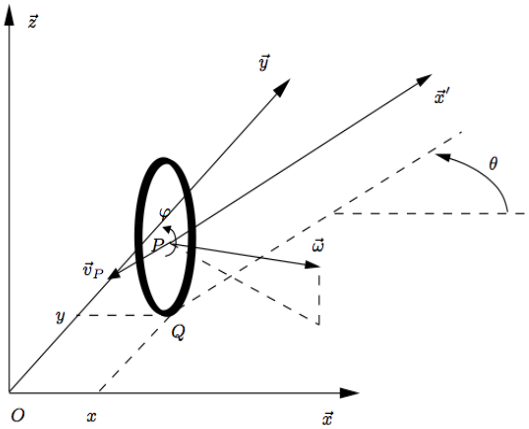

Figure 1: Characterization of rolling without slipping

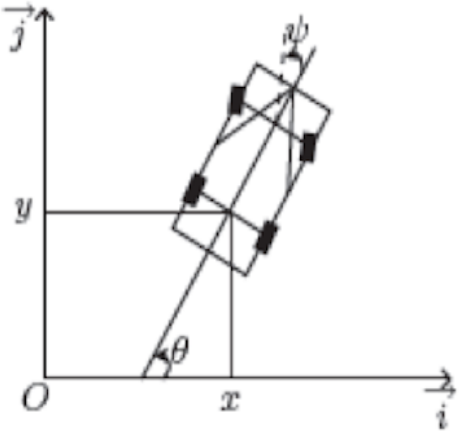

Figure 2: A robot of four wheels (car type)

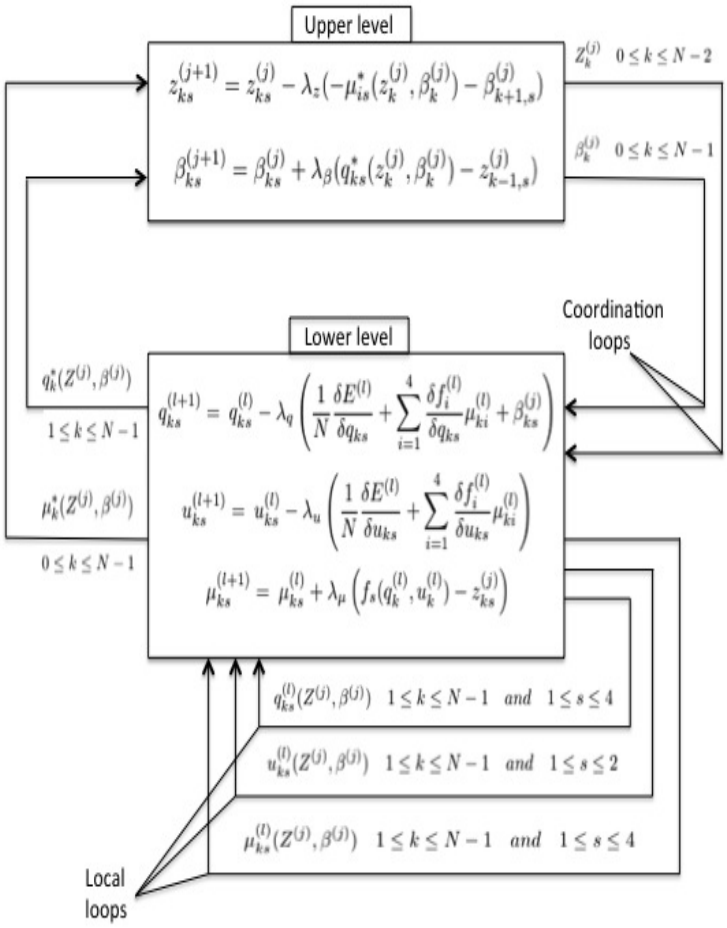

Figure 3: Coordination between Upper and Lower Level

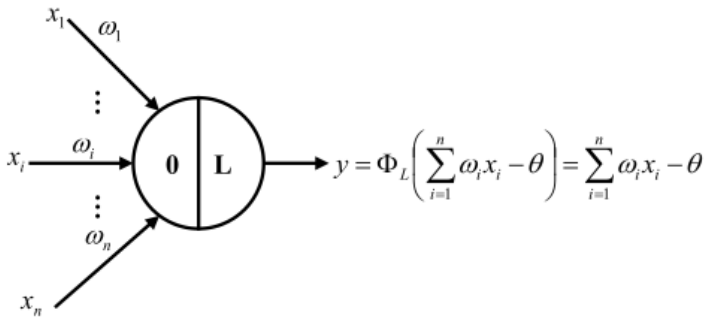

Figure 4: Linear Neuron

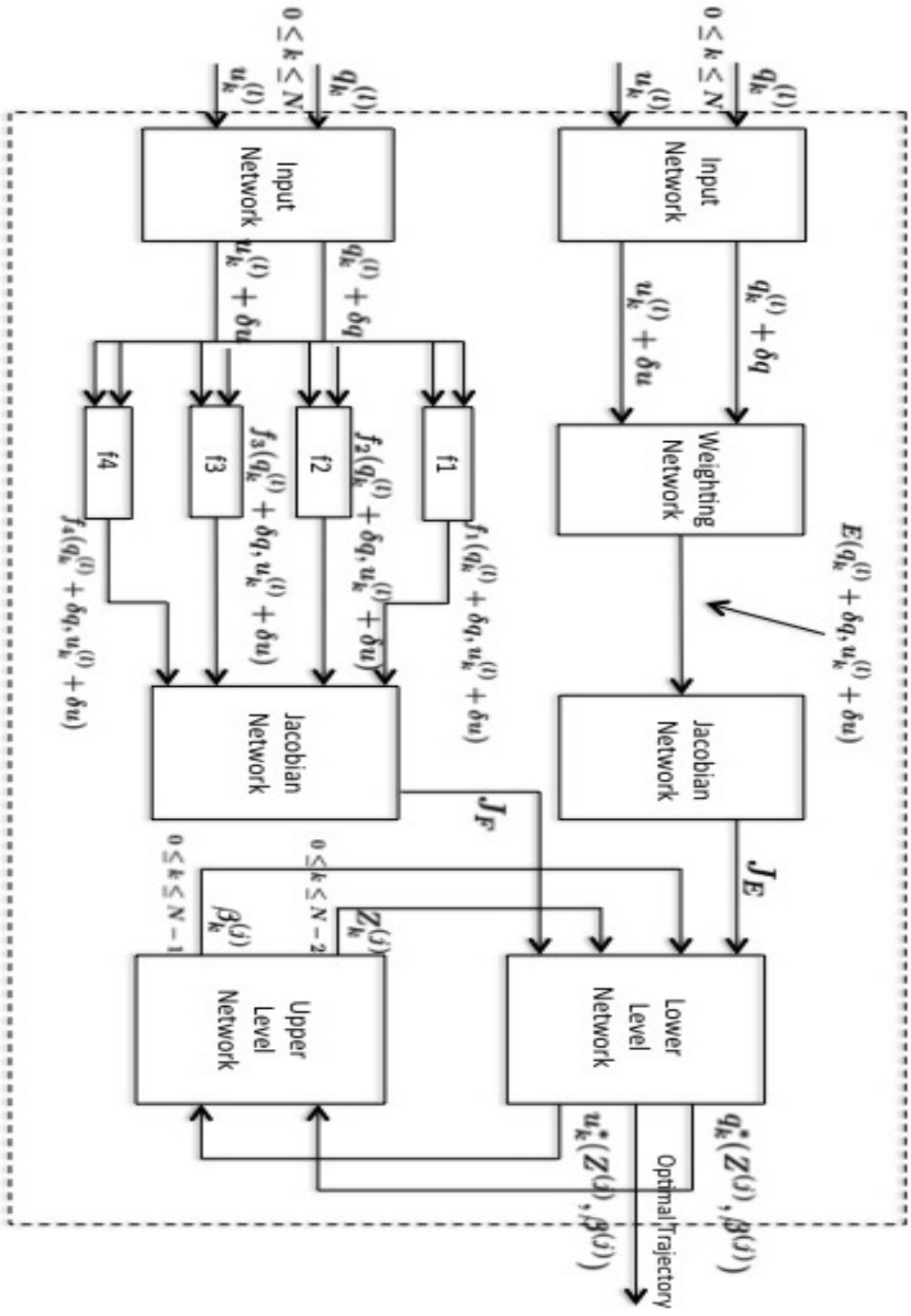

Figure 5: Architecture of the Final Network

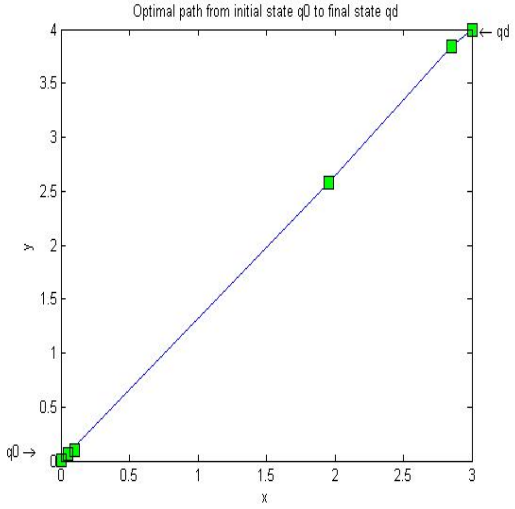

Figure 6: Optimal Trajectory-example 1

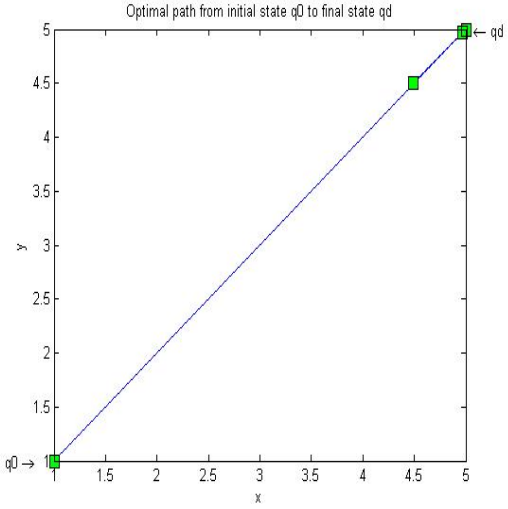

Figure 7: Optimal Trajectory-example 2

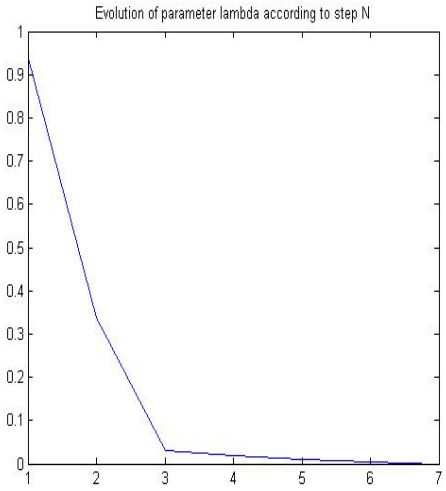

Figure 8: Evolution of Coordination parameter-example 1

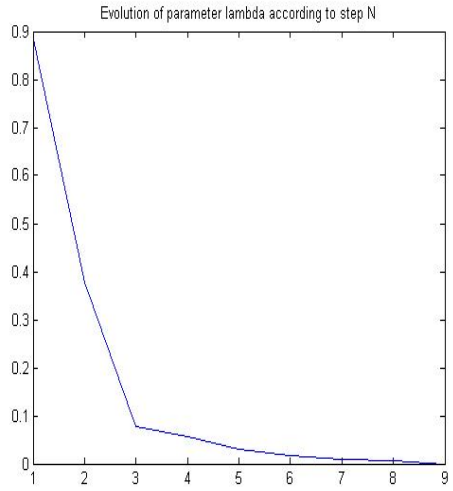

Figure 9: Evolution of Coordination parameter-example 2

References

[1] M.A. Abido, Multiobjective Evolutionary Algorithms for Electric Power Dispatch Problem, *IEEE Transactions on Evolutionary Computation*, **10** (2006), no. 3, 315-329. http://dx.doi.org/10.1109/tevc.2005.857073

[2] A. Abraham, L. Jain and R. Goldberg, Eds., *Evolutionary Multiobjective*

Optimization: Theoretical Advances and Applications, Springer-Verlag, 2005. http://dx.doi.org/10.1007/1-84628-137-7

[3] S.F. Adra, T.J. Dodd, I.A. Griffin and P.J. Fleming, Convergence Accelaration Operator for Multiobjective Optimization, *IEEE Transactions on Evolutionary Computation*, **13** (2009), no. 4, 825-847. http://dx.doi.org/10.1109/tevc.2008.2011743

[4] E. Aggelogiannaki and H. Sarimveis, A simulated annealing algorithm for prioritized multiobjective optimization implementation in an adaptive model predictive control configuration, *IEEE Transactions on Systems, Man and Cybernetics Part B-Cybernetics*, **37** (2007), no. 4, 902-915. http://dx.doi.org/10.1109/tsmcb.2007.896015

[5] S. Agrawal, B.K. Panigrahi and M.K. Tiwari, Multiobjective particle swarm algorithm with fuzzy clustering for electrical power dispatch, *IEEE Transactions on Evolutionary Computation*, **12** (2008), no. 5, 529-541. http://dx.doi.org/10.1109/tevc.2007.913121

[6] S. Agrawal, Y. Dashora, M.K. Tiwari and Y.J. Son, Interactive Particle Swarm: A Pareto-Adaptive Metaheuristic to Multiobjective Optimization, *IEEE Transactions on Systems, Man and Cybernetics Part A-Systems and Humans*, **38** (2008), no. 2, 258-277. http://dx.doi.org/10.1109/tsmca.2007.914767

[7] A.A. Aguilar-Lasserre, L. Piboleau, C. Azzaro-Pantel and S. Domenech, Enhanced genetic algorithm-based fuzzy multiobjective strategy to multiproduct batch plant design, *Applied Soft Computing*, **9** (2009), no. 4, 1321-1330. http://dx.doi.org/10.1016/j.asoc.2009.05.005

[8] H.E. Aguirre and K. Tanaka, Working principles, behavior and performance of MOEAs on MNK-landscapes, *European Journal of Operational Research*, **181** (2007), no. 3, 1670-1690. http://dx.doi.org/10.1016/j.ejor.2006.08.004

[9] C.W. Ahn, Ed., *Advances in Evolutionary Algorithms: Theory, Design and Practice*, Springer-Verlag, New York, USA, 2006. http://dx.doi.org/10.1007/3-540-31759-7

[10] K. Fujimura and H. Samet, A hierarchical strategy for path planning among moving obstacles (mobile robot), *IEEE Transactions on Robotics and automation*, **5** (2002), no. 1, 61-69. http://dx.doi.org/10.1109/70.88018

[11] S.S. Ge and Y.J. Cui, New Potential Functions for Mobile Robot Path Planning, *IEEE Transactions on Robotics and Automation*, **16** (2000), no. 5, 615-620. http://dx.doi.org/10.1109/70.880813

[12] H.W. Kuhn and A.W. Tucker, Nonlinear Programming, *Proceeding of the Second Berkeley Symposium on Mathematical Statistics and Probability*, 1950, 481-492. University of California Press, Berkeley and Los Angeles, 1951.

[13] M. Mestari, M. Benzirar, N. Saber and M. Khouil, Solving Nonlinear Equality Constrained Multiobjective Optimization Problems Using Neural Networks, *IEEE Transactions Neural Networks and Learning Systems*, **26** (2015), no. 10, 2500-2520. http://dx.doi.org/10.1109/tnnls.2015.2388511

[14] M. Mestari, An Analog Neural Network Implementation in Fixed Time of Adjustable-Order Statistic Filters and Applications, *IEEE Transactions on Neural Networks*, **15** (2004), no. 3, 766-785. http://dx.doi.org/10.1109/tnn.2003.820656

[15] M. Mestari, A. Namir, and J. Abouir, Switched capacitor neural networks for optimal control of nonlinear dynamic systems: Design and stability analysis, *Systems Analysis Modelling Simulation*, **41** (2001), no. 3, 559-591.

[16] A. Stenz, Optimal and efficient path planning for partially-known environments, *IEEE International Conference on Robotics and Automation*, (1994), 3310-3317. http://dx.doi.org/10.1109/robot.1994.351061

Received: November 2, 2015; Published: March 1, 2016