An Intelligent Compensation Through B-Spline Neural Network for a Delta Parallel Robot

Jonatan Martín Escorcia-Hernández, Hipólito Aguilar-Sierra, Omar Aguilar-Mejía, Ahmed Chemori, José Humbérto Arroyo-Núñez

An Intelligent Compensation Through B-Spline Neural Network for a Delta Parallel Robot

Jonatan Martín Escorcia-Hernández[1], Hipólito Aguilar-Sierra [1], Omar Aguilar-Mejía [2], Ahmed Chemori [3], and José Humbérto Arroyo-Núñez [1].

***Abstract*—In this paper a PD controller with intelligent compensation is used to solve the problem of tracking trajectories for a Delta Parallel Robot with three degrees of freedom. This controller uses an artificial B-Spline neural network as a feedforward compensation term. To evaluate the proposed controller performance some numerical simulations under two different scenarios have been carried out in order to know its effectiveness respect to a simple PD controller.**

***Index Terms*—Delta Parallel Robot, Learning systems, Trajectory tracking**

I. Introduction

The Delta Parallel Robot (DPR) was invented in the early 80's by Reymond Clavel (a professor at EPFL - Ecole Polytechnique Federale de Lausanne) [1]. This robot is mainly used in Pick and Place applications requiring high speed and good precision. The main industries where this type of robot is used are food, pharmaceutic, electronic, among others [2]. Devices based on DPR have also been used as 3D printers or haptic interfaces [1], [3].

For parallel robots, several control techniques have been implemented: nonlinear control, intelligent control, robust control, or a combination of the above to regulate the tracking of complex trajectories [4]. The Proportional-Integral-Derivatives controllers (PID) has been widely used in the industry for its simplicity and good performance [5]. However, the performance of the PID controller on DPR decreases due to internal disturbances caused by the set of closed kinematic chains [6], [7]. One of the main causes of the poor performance of the PID controller is due to the controller gains being calculated arbitrarily or by trial and error. The computed torque control is another control technique regularly used in parallel robots; this technique is based on inverse dynamics, so it is necessary to have knowledge of the matrices that make up the dynamics of the robot, [8], [9]. In [6] a fuzzy controller is used to regulate trajectory tracking of a DPR; in this controller the parameters of the fuzzy controller are adjusted by a particle swarm optimization algorithm. Although the results based on the fuzzy logic controller present a good performance in a closed loop, many problems related with the fuzzification rules exist, for example the defuzzification operations are unclear; besides this, operations demand high computational processing capacity. Due to the complexity of the dynamic model of DPR, some terms of the dynamic model are unknown, whereby control techniques are used based on state observers. In [7] an active disturbance rejection control (ADRC) is used, where the unknown terms are estimated by high gain observers. The ADRC cancels the effects produced by internal or external disturbances in the DPR and compensates the effects produced by unknown dynamics to decrease steady-state error. However the accuracy estimation of unknown parameters depends on a good tuning process at the gains [7], [10]. In [11] a hybrid controller is used based on two controllers that work in parallel. The first controller has the function of regulating the joint acceleration of DPR through a disturbance observer. The second controller is a sliding mode controller, which imposes the desired dynamic of the tangential, normal and bi-normal components of the tracking error signal, with the purpose of reducing the path tracking contour error. A drawback that the variable structure controllers show is the effects in the response, which is due to high switching frequencies.

The artificial neural networks (ANN) are used in the automatic control area to recognize parameters of nonlinear systems, adaptive control systems design, and intellient compensators [12]. Since the end of the 20th century, the use of ANN has been increasing in several areas of knowledge due to its great capacity to adapt to various engineering problems and industrial applications [13]. Controllers that use ANN are employed to regulate nonlinear systems due to parametric uncertainty, unknown dynamic, and high coupling between state variables in the mathematical model of the system [14]. Artificial neural networks of instant learning exist, such as B-Spline Neural Networks (BSNN), that have favorable aspects and characteristics compared with other intelligent networks such as the back-propagation (BP) ANN and the radial base function (RBF) ANN [15]. The main advantage of the BSNN with respect to another ANN is that the control law is adapted every moment in the online process, while the other types need a previous offline training. The drawback of offline training is if the operating condition changes, it is necessary do another offline training again [16]. BSNN can be defined as a system that converts input patterns to their corresponding weight without performing many operations. It should be mentioned that the ANN BP and RBF need an offline training before implementing in the experiment or prototype. This situation requires more execution

[1] J. Escorcia, H. Aguilar, and J. Arroyo are with Universidad Politécnica de Tulancingo, Calle Ingenierías No. 100 C.P. 43629 Tulancingo Hidalgo México. Email: {1715002,hipolito.aguilar,humberto.arroyo}@upt.edu.mx

[2] O. Aguilar is with Universidad Popular Autónoma del Estado de Puebla, Departamento de Posgrado, C.P. 72410. Email: omar.aguilar@upaep.mx

[3] A. Chemori is with LIRMM, Université de Montpellier - CNRS, 161 rue Ada, 34090 Montpellier France. Email: Ahmed.Chemori@lirmm.fr

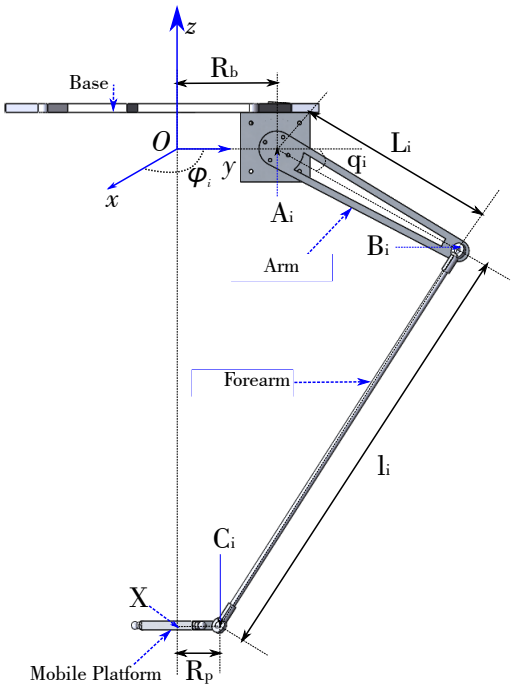

Fig. 1. DPR geometric schematic diagram

time and more number of operations, and sometimes it does not converge to a solution for the system. Whereby the BSNN is a great alternative for robust control applications, adaptive filters, real-time control, nonlinear systems modeling, pattern recognition, among others. With the training strategy online, it ensures the modification of the weights of the ANN so that the DPR follows the path under different operating conditions.

In this paper a control scheme based on a BSNN of instant training is proposed to regulate the trajectory tracking of a DPR. The BSNN is used as a feedforward compensation term in order to reduce the trajectory tracking error for a Pick and Place task.

The organization of this paper is as follows: In Section II the inverse dynamic model of DPR is presented. In Section III the proposed PD controller with BSNN compensation is described in detail. Section IV describes the simulation results of the PD BSNN controller whose performance is compared versus the performance of a simple PD controller. Finally the conclusions of this paper are given in Section V.

II. Dynamic Model of the DPR

The DPR consists essentially of two platforms, the base or fixed platform and the mobile platform; both are joined by three kinematic chains, each kinematic chain consists of two parts, the arm and the forearm. The robot arms are mounted directly to the actuators in the fixed platform through rotational joints. The robot forearms consist of two parallel bars, which connect the arm with the mobile platform via ball joints; the end-effector is located on the mobile platform. The dynamic model is represented in the joint space whose variables are denoted as $q = [q_1 \quad q_2 \quad q_3]^T$. The schematic diagram of a DPR kinematic chain is show in Fig. 1. The inverse dynamic model for the DPR has been taken by the work developed by [17]. Some simplifications for the dynamic model have been taken into account, these simplifications are discussed in more detail in these works [18], [19].

- The frictional forces of all types, whether dry or viscous, are neglected.
- The forearms mass are smaller than other DPR parts, its inertia can be neglected hence, forearm mass is divided in to two parts, one is added to DPR arm mass and the other part is added to the mobile platform mass.

Two different forces can be distinguished acting on the mobile platform. The gravity force which can be expressed by:

$$G_P = -M_p G \tag{1}$$

where $M_p = diag([m_t \quad m_t \quad m_t])$ with $m_t = m_p + 3\frac{m_{fa}}{2}$, m_p is the mobile platform mass, and m_{fa} is the forearm mass. The gravity vector $G \in \mathbb{R}^{3\times1}$ is given by $G = [0 \quad 0 \quad g]^T$ with $g = 9.81\ m/s^2$.

The inertial forces on the mobile platform due to the Cartesian acceleration $\ddot{X} \in \mathbb{R}^{3\times1}$ are:

$$F_p = M_p \ddot{X} \tag{2}$$

The torque contributions of G_p and F_p to each motor located in the base denoted by T_{Gp} and T_{Fp} can be calculated using the inverse Jacobian matrix $J_{inv}(q, X) \in \mathbb{R}^{3\times3}$

$$T_{Gp} = -J_{inv}^T M_P G \tag{3}$$

$$T_{Fp} = J_{inv}^T M_p \ddot{X} \tag{4}$$

The acting forces on the DPR arms are the torque produced by the motors $\tau \in \mathbb{R}^{3\times1}$, the torque due to the acceleration of the arms $T_{AA} \in \mathbb{R}^{3\times1}$ and torque due to gravity on the arms $T_{AG} \in \mathbb{R}^{3\times1}$; each contribution is represented as follows:

$$T_{AA} = I_{AA} \ddot{q} \tag{5}$$

$I_{AA} \in \mathbb{R}^{3\times3}$ is a diagonal matrix whose elements of the diagonal are formed by:

$$I_{aa} = I_{act} + I_{arm} + \frac{L_i^2 m_f}{2} \tag{6}$$

Where I_{act}, I_{arm} are the inertia of the actuators and the inertia of the arms, respectively and L_i is the arm length. The torques produced by the gravitational forces acting on the arms are given by

$$T_{AG} = -M_{Ra} g \cos(q) \tag{7}$$

$$M_{Ra} = diag([m_{ra} \quad m_{ra} \quad m_{ra}]) \tag{8}$$

$$\cos(q) = [\cos(q_1) \quad \cos(q_2) \quad \cos(q_3)]^T \tag{9}$$

Where $m_{ra} = m_a L_c + \frac{m_{fa} L}{2}$, L_c is the length to the center of mass of one arm, and m_{fa} is the mass of the forearm.

Applying the virtual work principle, which states that the sum of all non-inertial forces must be equal to the sum of all inertial ones, we obtain:

$$\tau - T_{Gp} - T_{AG} = T_{AA} + T_{Fp} \tag{10}$$

TABLE I
DPR GEOMETRIC PARAMETERS

Parameter	Description	Value
L	Arm length	0.3 m
l	Forearm length	0.624 m
R_b	Base platform radio	0.1267 m
R_p	Mobile platform radio	0.0497 m

TABLE II
DPR DYNAMIC PARAMETERS

Parameter	Description	Value
m_p	Mobile platform mass	0.19 Kg
m_a	Arm mass	0.29 Kg
m_{fa}	Forearm mass	0.28 Kg
I_{arm}	Arm inertia	0.0213 Kgm^2
I_{act}	Motor inertia	3.8 $\times 10^{-6}$ Kgm^2

To express only the dynamic model in the joint space, we proceed to use the following relationship:

$$\ddot{X} = J_{inv}\ddot{q} + \dot{J}_{inv}\dot{q} \tag{11}$$

Rearranging the terms, the inverse dynamic model of the DPR can be written into the standard joint space form as follows:

$$M(q)\ddot{q} + C(q,\dot{q})\dot{q} + G(q) = \tau \tag{12}$$

where:

- $M(q) = I_{AA} + J_{inv}^T M_P J_{inv}$
- $C(q,\dot{q}) = J_{inv}^T M_P \dot{J}_{inv}$
- $G(q) = T_{GP} + T_{AG}$

The DPR kinematic and dynamic parameters are shown in Tables 1 and 2 respectively.

III. CONTROL STRATEGY

The control law for the DPR is given as follows:

$$\tau = K_p e_q(t) + K_d \dot{e}_q(t) + \hat{\sigma}(e_q) \tag{13}$$

where $K_p \in \mathbb{R}^{3\times3}$, $K_d \in \mathbb{R}^{3\times3}$ are diagonal positive definite matrices which are the feedback gains proportional and derivative respectively; the joint tracking error expression is given by $e_q(t) = q_d(t) - q(t)$ where $q_d(t) \in \mathbb{R}^{3\times1}$ and $q(t) \in \mathbb{R}^{3\times1}$ are the desired joint trajectory and the measured joint position respectively.

The parts that integrate the structure of the BSNN are divided in three main parts, defined as follows: A l-dimensional space to normalized inputs, a set of base functions, and the output function of the BSNN. Fig. 2 shows the main components that make up the BSNN and Fig. 3 shows the whole DPR control diagram. The main elements in the BSNN structure are the base functions, which are specified from a set of control point vectors. The Base functions of the BSNN are obtained using a periodic expression described by different authors [20], [21]; these functions are numerically stable, computationally efficient, and can deal with any strategic distribution of control points which are determined in an off-line process and must be delimited according to the possible BSNNs input values.

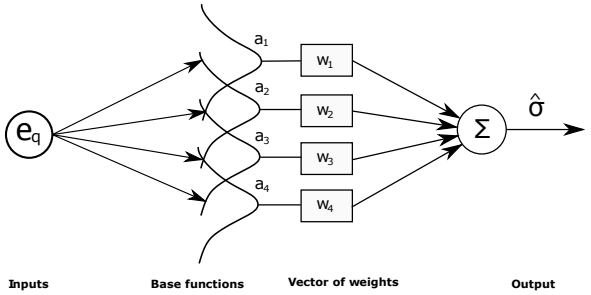

Fig. 2. Diagram of the proposed BSNN used as a compensation term for a PD controller

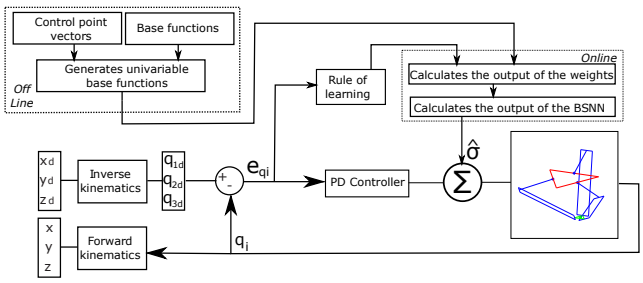

Fig. 3. DPR control scheme with BSNN compensation

The output of the BSNN is obtained through a linear combination of the base functions' outputs. Another important aspect of the BSNN is that the instant training is simply a linear optimization problem because the adjustable weights (w_i) are linear coefficients, which makes the output linearly dependent on the set of weights [22]. The output of the BSNN can be written as [23]:

$$\hat{\sigma}_i = \sum_{i=1}^{P} a_i w_i = \mathbf{a}^T \mathbf{w} \tag{14}$$

where $\mathbf{a}$ is a P-dimensional vector which contains the outputs of the base for $P = 1, ..., 4$ and, $\mathbf{w}$ is the weights vector. The B-Spline base function is defined in the following form [24]:

$$\begin{aligned} a_K^j(e_q) = &\left(\frac{e - \lambda_{j-K}}{\lambda_{j-1} - \lambda_{j-K}}\right) a_{K-1}^{j-1}(e_q) + \\ &\left(\frac{\lambda_j - e_q}{\lambda_j - \lambda_{j-K+1}}\right) a_{K-1}^{j}(e) \\ a_1^j(e_q) = &\begin{cases} 1 & \text{if } e_q \in I_j \\ 0 & \text{other case} \end{cases} \end{aligned} \tag{15}$$

where λ_j is the j-th control point and $I_j = [\lambda_{j-1}, \quad \lambda_j)$ is the j-th interval for $j = 1, ...4$, and K is the B-Spline function order which can take values of 1 to 4. The values for each control point λ_j are defined according to the range of possible values of the error signal.

A. Learning rule

The performance function establishes the type of learning rule, computational complexity and final model. In this paper a simple learning rule is required for its implementation, whereby a performance function of medium quadratic error (MQE) is

selected because it provides excellent results in most cases [15]. The rules of instant learning are formulated by minimizing the instantaneous estimation of a MQE output performance function and the parameters are upgraded using the downward gradient rules. The gradient method can be implemented in two different ways: a) batch learning and b) online learning. Batch learning corresponds to the standard gradient method, where the network weights are updated only once in each iteration of the training process, after all learning examples are procesed by the network [25]. The online training is a variation of the standard gradient method, where the networks weights are upgraded after every learning example is processed. For this case we chose to implement the online learning method. In neural networks computational engineering, the gradient method is commonly employed due to its simplicity and efficiency [26]. In the neuro-controller design, rules of downward gradient are used, and the updating of the weights $\Delta\mathbf{W}(t-1)$ is done by means of an instant learning rule, as follows [23], [27]:

$$\Delta\mathbf{W}(t-1) = \frac{\gamma\tilde{\sigma}(t)}{||\mathbf{a}(t)||_2^2}\mathbf{a}(t) \tag{16}$$

where γ is the learning relationship, $\mathbf{a}$ is the vector that contains the output of the base functions, $\mathbf{W}$ is the weights vector, and $\tilde{\sigma}(t) = \sigma(t) - \hat{\sigma}(t)$, where $\sigma(t)$ and $\hat{\sigma}(t)$ are the actual output and the desired output of the BSNN respectively.

IV. Simulation and Results

The performance of the proposed PD controller with BSNN compensation is evaluated using a trajectory for a Pick and Place task. This trajectory is generated using polynomial interpolation of degree five [28], [29]. The function is given in this form:

$$x_f = x_i + r(t)\Delta x, \qquad for \quad 0 \le t \le t_f \tag{17}$$

And:

$$r(t) = 10\left(\frac{t}{t_f}\right)^3 - 15\left(\frac{t}{t_f}\right)^4 + 6\left(\frac{t}{t_f}\right)^5 \tag{18}$$

Where x_i is the initial position, x_f is the final position, both are given in cartesian space, $r(t)$ is the trajectory function of two points, $\Delta_x = x_f - x_i$, and t_f is the duration of the movement. Using (17) and (18) the desired trajectory for a Pick and Place Task is generated which is shown in Fig. 4. For the DPR simulation, the system is submitted to two different scenarios. For the first scenario, DPR executes the trajectory shown in Fig. 4 without any load on the mobile platform. In the second scenario DPR executes the same trajectory but a load of 1 Kg is added at different intervals of the trajectory; Fig. 5 shows the desired trajectory in 3D with the changes in the mass used for scenario 2.

The proposed PD with BSNN compensation performance is compared with a simple PD controller in order to evaluate the improvement obtained when a compensation term is added. The parameters for the controllers are summarized in Table 3. To quantify the performance of the proposed PD controller with BSNN compensation versus the simple PD controller the Root Mean Square Error (RMSE) formula is used, which allows us to know with better precision which of the proposed controllers offers a better performance for tracking trajectory tasks. The

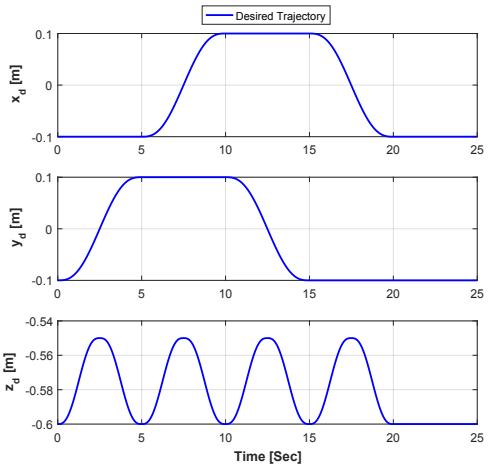

Fig. 4. Desired trajectory for a Pick and Place Task

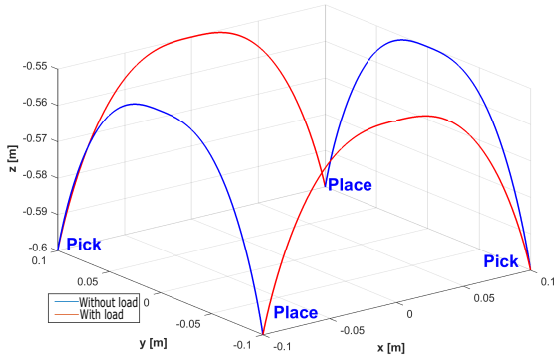

Fig. 5. 3D Trajectory for a Pick and Place Task. The lines in red correspond to the section of the trajectory with load, and the blue lines without load

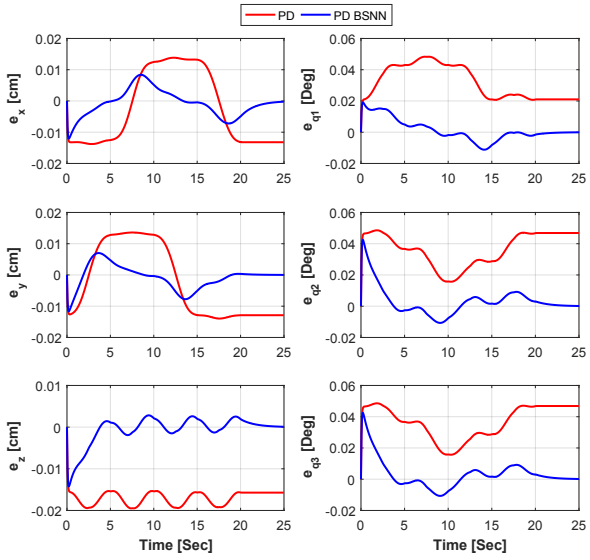

Fig. 6. The first column corresponds to tracking error in Cartesian Space, and the second one in Joint Space Scenario 1

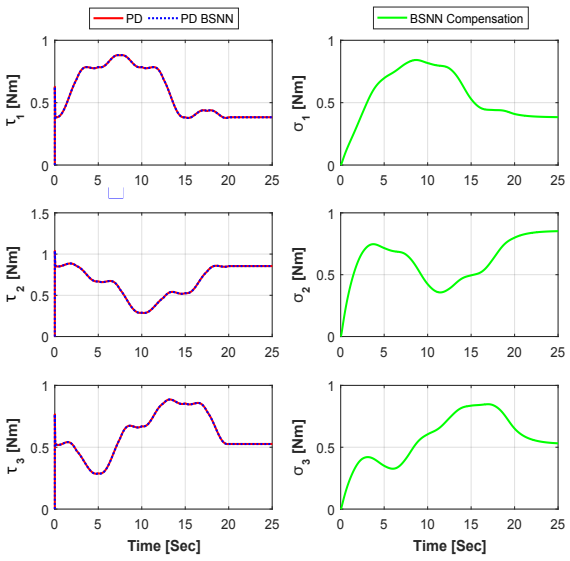

Fig. 7. The first column are the control signals, and the second one to BSNN compensation term Scenario 1

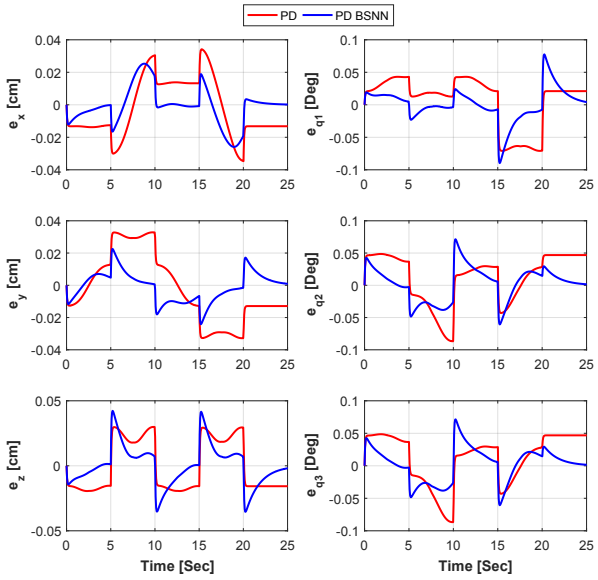

Fig. 8. The first column corresponds to tracking error in Cartesian Space, and the second one in Joint Space Scenario 2

TABLE III
CONTROLLERS PARAMETERS

PD/PD BSNN
$K_P = 1045$
$K_D = 63$
$\gamma = 0.53$
$I_1 = [-1.5 \quad -1.2 \quad -0.9 \quad -0.6]$
$I_2 = [-0.9 \quad -0.6 \quad -0.3 \quad 0]$
$I_3 = [-0.3 \quad 0 \quad 0.3 \quad 0.6]$
$I_4 = [0.3 \quad 0.6 \quad 0.9 \quad 1.2]$

RMSE for the Cartesian and joint space are given as follows respectively:

$$RMSE_C = \sqrt{\frac{1}{N}\sum_{k=1}^{N}(e_x^2(k) + e_y^2(k) + e_z^2(k))} \tag{19}$$

$$RMSE_J = \sqrt{\frac{1}{N}\sum_{k=1}^{N}(e_{q1}^2(k) + e_{q2}^2(k) + e_{q3}^2(k) + e_{q4}^2(k))} \tag{20}$$

where e_x, e_y, e_z denote the Cartesian position tracking error of the mobile platform along the x, y, z axes, while e_{q1}, e_{q2}, e_{q3} are the different joint space tracking errors. Moreover, N is the number of samples and k the sample at a certain moment. The RMSE results for scenarios 1 and 2 are depicted in Tables IV and V respectively. According to the results, the improvement of the proposed PD BSNN controller exceeds the simple PD controller more than 70% for the first scenario when the DPR woks without any payload; likewise for scenario 2, the improvement of the PD BSNN controller is greater than 30% with respect to the PD controller. The control signals with respect to time for both controllers and the behavior of the BSNN compensation term with respect to time are illustrated in Fig. 7 for scenario 1 and in Fig. 9 for scenario 2. As can be noted in Fig. 6 and Fig. 8 the steady-state error for the PD

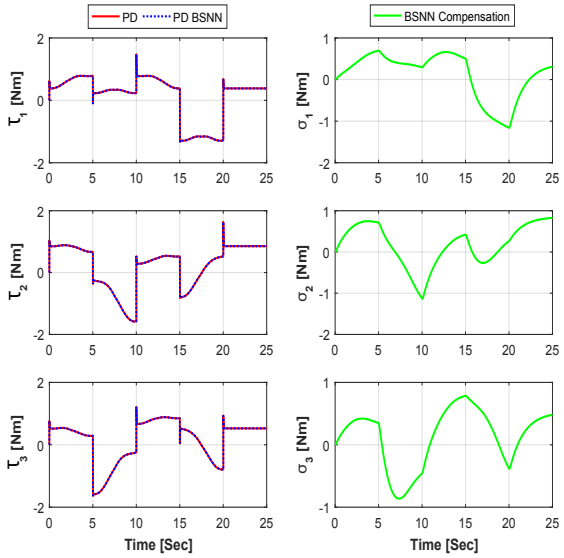

Fig. 9. The first column are the control signals, and the second one to BSNN compensation term Scenario 2

controller will never be zero; this is because the PD controller can not compensate the gravity term of the manipulator, unless the gravity vector is known and it can be include into the controller, or add an integral action at the PD controller, unlike the PD BSNN controller which, due to the neural network, tries to estimate the DPR dynamic model and in this way cancels the effects produced by gravity. PD BSNN controller alleviates the steady-state error in 3 seconds for the first scenario and 5 seconds for the second scenario.

V. CONCLUSIONS AND FUTURE WORK

In this work we proposed a PD controller with intelligent compensation based on the B-Spline neural network which was applied in simulation to a DPR. An advantage of this

TABLE IV
CONTROLLERS PERFORMANCE EVALUATION SCENARIO 1

	$\mathbf{RMSE_C}$ $(\mathbf{cm})$	Cartesian Improvement	$\mathbf{RMSE_J}$ $(\mathbf{Deg})$	Articular Improvement
PD	0.0239	0 %	0.0614	0 %
PD BSNN	0.0067	71.81 %	0.0138	77.61 %

TABLE V
CONTROLLERS PERFORMANCE EVALUATION SCENARIO 2

	$\mathbf{RMSE_C}$ $(\mathbf{cm})$	Cartesian Improvement	$\mathbf{RMSE_J}$ $(\mathbf{Deg})$	Articular Improvement
PD	0.0343	0 %	0.0674	0 %
PD BSNN	0.0214	37.67 %	0.0450	33.16 %

controller is that it is not necessary to have knowledge about the dynamic parameters of the system, only the range of possible values that the error signal can acquire should be considered. The results show the improvement of including the term of intelligent compensation, since it considerably improved the performance of the system under different requirements. For future work it is intended to implement this controller in real-time to a physical DPR and compare the performance obtained with the BSNN with respect to a PD controller with Radial Base Function Neural Network compensation.

ACKNOWLEDGMENTS

Thanks to the Program of Support to the development of higher education (PADES), agreement No 2018-13-011-047. Additionally J. Escorcia thanks to The Mexican Council of Science and Technology (CONACYT); Award no. 593804.

REFERENCES

[1] R. Clavel, "Delta, a fast robot with parallel geometry," in *Proc. 18th Int. Symp. on Industrial Robots, Lausanne, 1988*, 1988, pp. 91–100.

[2] J. Brinker and B. Corves, "A survey on parallel robots with delta-like architecture," in *Proceedings of the 14th IFToMM World Congress*, 2015, pp. 407–414.

[3] H. D. Taghirad, *Parallel robots: mechanics and control*. CRC press, 2013.

[4] G. Sartori Natal, "Control of parallel robots: towards very high accelerations," Ph.D. dissertation, Université Montpellier 2, 2012.

[5] K. Ogata and Y. Yang, *Modern control engineering*. Prentice-Hall Englewood Cliffs, NJ, 2009.

[6] X. Lu and M. Liu, "A fuzzy logic controller tuned with pso for delta robot trajectory control," in *Industrial Electronics Society, IECON 2015-41st Annual Conference of the IEEE*. IEEE, 2015, pp. 004 345–004 351.

[7] L. A. Castañeda, A. Luviano-Juárez, and I. Chairez, "Robust trajectory tracking of a delta robot through adaptive active disturbance rejection control," *IEEE Transactions on Control Systems Technology*, vol. 23, no. 4, pp. 1387–1398, 2015.

[8] R. Kelly, V. S. Davila, and J. A. L. Perez, *Control of robot manipulators in joint space*. Springer Science & Business Media, 2006.

[9] M. W. Spong, S. Hutchinson, and M. Vidyasagar, *Robot modeling and control*. wiley New York, 2006, vol. 3.

[10] H. Sira-Ramírez, J. Linares-Flores, A. Luviano-Juarez, and J. Cortés-Romero, "Ultramodelos globales y el control por rechazo activo de perturbaciones en sistemas no lineales diferencialmente planos," *Revista Iberoamericana de Automática e Informática Industrial RIAI*, vol. 12, no. 2, pp. 133–144, 2015.

[11] T. Uzunovic, E. A. Baran, E. Golubovic, and A. Sabanovic, "A novel hybrid contouring control method for 3-dof robotic manipulators," *Mechatronics*, vol. 40, pp. 178–193, 2016.

[12] W. Yu, *PID Control with Intelligent Compensation for Exoskeleton Robots*. Academic Press, 2018.

[13] H. Wang and Q. Zou, "B-spline-decomposition-based approach to multi-axis trajectory tracking: Nanomanipulation example," *IEEE Transactions on Control Systems Technology*, vol. 22, no. 4, pp. 1573–1580, 2014.

[14] H. Deng, D. Srinivasan, and R. Oruganti, "A b-spline network based neural controller for power electronic applications," *Neurocomputing*, vol. 73, no. 4, pp. 593–601, 2010.

[15] O. Aguilar, R. Tapia, A. Valderrabano, and H. Minor, "Design and performance comparison of pi and adaptive current controllers for a wecs," *IEEE Latin America Transactions*, vol. 13, no. 5, pp. 1361–1368, 2015.

[16] K. Cheng, H. Wang, and D. Sutanto, "Adaptive directive neural network control for three-phase ac/dc pwm converter," *IEE Proceedings-Electric Power Applications*, vol. 148, no. 5, pp. 425–430, 2001.

[17] M. Bennehar, A. Chemori, and F. Pierrot, "A novel rise-based adaptive feedforward controller for redundantly actuated parallel manipulators," in *Intelligent Robots and Systems (IROS 2014), 2014 IEEE/RSJ International Conference on*. IEEE, 2014, pp. 2389–2394.

[18] D. Corbel, M. Gouttefarde, O. Company, and F. Pierrot, "Towards 100g with pkm. is actuation redundancy a good solution for pick-and-place?" in *2010 IEEE International Conference on Robotics and Automation*. IEEE, 2010, pp. 4675–4682.

[19] F. Pierrot, C. Reynaud, and A. Fournier, "Delta: a simple and efficient parallel robot," *Robotica*, vol. 8, no. 2, pp. 105–109, 1990.

[20] M. G. Cox, "The numerical evaluation of b-splines," *IMA Journal of Applied Mathematics*, vol. 10, no. 2, pp. 134–149, 1972.

[21] C. De Boor, "On calculating with b-splines," *Journal of Approximation Theory*, vol. 6, no. 1, pp. 50–62, 1972.

[22] S. S. Rao and S. S. Rao, *Engineering optimization: theory and practice*. John Wiley & Sons, 2009.

[23] M. Brown and C. J. Harris, "Neurofuzzy adaptive modelling and control," 1994.

[24] L. Mirea, "Dynamic multivariate b-spline neural network design using orthogonal least squares algorithm for non-linear system identification," in *System Theory, Control and Computing (ICSTCC), 2014 18th International Conference*. IEEE, 2014, pp. 720–725.

[25] R. T. Olvera, "Utilización del neurocontrolador b-spline para regular el statcom," Ph.D. dissertation, Centro de Investigación y de Estudios Avanzados del I.P.N. Unidad Guadalajara, 2006.

[26] L. dos Santos Coelho and M. W. Pessôa, "Nonlinear identification using a b-spline neural network and chaotic immune approaches," *Mechanical Systems and Signal Processing*, vol. 23, no. 8, pp. 2418–2434, 2009.

[27] J. M. Ramirez and O. Ruben Tapia, "Neural network control of the statcom in multimachine power systems," *WSEAS Transaction on PowerSystems*, vol. 2, pp. 1790–5060, 2007.

[28] W. Khalil and E. Dombre, *Modeling, identification and control of robots*. Butterworth-Heinemann, 2004.

[29] G. S. Natal, A. Chemori, and F. Pierrot, "Dual-space control of extremely fast parallel manipulators: payload changes and the 100g experiment," *IEEE Transactions on Control Systems Technology*, vol. 23, no. 4, pp. 1520–1535, 2015.