A Comparative Study of Optimal First and Higher Order Sliding Mode Control for a Robot Manipulator

Mohammad Hashemzadeh, MM. Fateh, and M. Hadad Zarif

***Abstract*— This paper presents a comparative study of Higher Order Sliding Mode (HOSM) Control and first order Sliding Mode which is simply known as Sliding Mode Control (SMC). This comparative study capacitates one to observe performance qualities of both controllers. First, the procedures of designing of both techniques are presented. After that for a better comparison, both control approaches are optimized using Particle Swarm Optimization. Finally, simulation results will be showed for the nominal model of a robot manipulator which reveal the merits and demerits of both methods. A desired path is defined as a tracking task. Although simulation results show an overall improvement for HOSM over SMC, the latter still displays some good points over the former.**

I. Introduction

Undoubtedly, Sliding Mode Control, SMC [1], is one of the most frequent and well-liked techniques in the control of systems under heavy uncertainty. This method is based on maintaining the system in hand close enough to a desired hyperplane called *Sliding Surface*, σ, which is suitably designed to achieve several control goals such as stability and tracking. The implementation of SMC can be done by high-frequency switching between some control signal's values, which results in unfavorable chattering phenomenon that is usually extremely dangerous, as it may excite the higher order frequencies of the system.

Having studied the literature, one can find several methods to mitigate the chattering problem. An author [1] proposes a time-varying switching gain, which does not need a new dynamical design. Another method [2] is to use a multi-phase SMC. The chattering event can also be alleviated by using an observer [3]. Slotine [4] suggested a boundary layer which means switching from discontinuous to continuous control.

Having preserved all the good points of SMC, Higher Order Sliding Mode (HOSM) [5-6] is a novel approach that can eliminate the chattering effect. Instead of using the first derivative of σ, one can use the *r-th* derivative of the sliding surface, which *r* is the relative degree [4] of the system. This technique also has a better degree of accuracy regarding the sampling time than the standard SMC. Although there are a great number of articles for SISO systems, HOSM is not easily applicable for MIMO ones. Fortunately, second order sliding mode can be found in some papers [7-8], and the algorithm for designing HOSM for a MIMO system has recently published [9].

The aim of this article is to compare standard SMC and HOSM for a robot manipulator. PUMA 560 [10] is quite a common robot in both industry and academic laboratories. To draw a better comparison, both control methods are optimized by Particle Swarm Optimization (PSO) [11], which is an evolutionary algorithm. Merits and demerits of both methods are presented.

The organization of the rest of the paper is as follows: section II presents the robot dynamics, section III is about control design, section IV talks about PSO, section V includes simulation results, and finally, section VI dedicates to the conclusion.

II. Robot Dynamics and Kinematics

Basically, robots [12] have three parts: mechanical, electrical and control. Mechanical parts are links between the base and the end-effector. Electrical parts are for moving or rotating the mechanical parts, and last but not least, control parts are for timing and organizing the other pars. As mentioned before, PUMA 560 is a very popular robot in both industrial and academic places. The mathematical model as well as parameters is included in several papers, e.g. [10].

A. Robot dynamics

Dynamical equation [10] for robots is as (1):

$$M(q)\ddot{q} + N(q,\dot{q}) = \tau \tag{1}$$

τ, 3x1, is control input vector, q is the joint vector that is joints angular position vector, $M(q)$, 3x3, is inertia matrix and $N(q,\dot{q})$, 3x1, is a vector including nonlinear parts which refers to gravity force and Coriolis and centrifugal terms.

B. Forward and inverse kinematics

Forward kinematics deals with the position of the end-effector when the joints angles are known. Inverse kinematics on the other hand, tries to find angles between the joints while position of the end-effector is available. In order to remain in the valid region of values for robot variables, first, one can use the inverse kinematics [13] for the desired path of the robot and then, calculate the control law according to those values. Hence, the reference angles would be always in the valid intervals.

Mohammad Hashemzadeh is with the Shahrood University of Technology, Shahrood, Iran (corresponding author to provide phone: +98-913-104-5365 e-mail: s.h.hashemzadeh@ieee.org).

MM. Fateh is with the Department of Electrical and Robotic Engineering, Shahrood University of Technology, Shahrood, Iran (e-mail: mmfateh@shahroodut.ac.ir).

M. Hadad Zarif is with the Department of Electrical and Robotic Engineering, Shahrood University of Technology, Shahrood, Iran (e-mail: mhzarif@shahroodut.ac.ir).

III. Controller Design

The objective of the controllers in this article is tracking control in the presence of uncertainty. Torque control of PUMA 560, which is a nonlinear uncertain Multi-Input-Multi-Output system with a wide range of operation, can be done using Robust Control methods such as SMC [4] or HOSM [9], which their stability has been proved based on Lyapunov direct method. Robot parameters are included in [10].

A. Standard SMC design for a MIMO system

A standard SMC design is presented in [4]. To adapt the equations for PUMA 560, one can choose only 3 states for state equations. State equations are:

$$\ddot{X} = F + BU \ , \ F = \begin{bmatrix} f_1 \\ f_2 \\ f_3 \end{bmatrix} \ , \ U = \begin{bmatrix} \tau_1 \\ \tau_2 \\ \tau_3 \end{bmatrix} \ , \ \ddot{X} = \ddot{q} \tag{2}$$

U is the control input, $\ddot{q}$ is as in (1), F is a term for nonlinear parts, B is control signal's coefficients, and also:

$$B = (I + \Delta)\hat{B} \ , \ |\Delta_{ij}| \le D_{ij} \ , \ |f_i - \hat{f}_i| \le F_i \ , i, j = 1, 2, 3 \tag{3}$$

$\hat{B}$ is the matrix of nominal values of matrix B, and upper bounds of uncertainties are F_i and D_{ij}.

Regarding (1) and (2), for a robot, one can write:

$$B = \hat{M}^{-1} \ , \ F = \hat{M}^{-1} * N(q, \dot{q}) \tag{4}$$

The sliding surface is as follows:

$$s_i = (\frac{d}{dt} + \lambda_i)\tilde{x}_i = \dot{\tilde{x}}_i + \lambda_i \tilde{x}_i \ , i = 1, 2, 3 \tag{5}$$

where λ_1, λ_2 and λ_3 are design parameters, $\tilde{x}_1, \tilde{x}_2, \tilde{x}_3$ are state errors, and sliding conditions are:

$$\frac{1}{2}\frac{d}{dt}(s_i^2) \le -\eta_i * \operatorname{sgn}(s_i) \ , \ i = 1, 2, 3$$

$$\Rightarrow \dot{s}_i \le -\eta_i * \operatorname{sgn}(s_i) \ , \ i = 1, 2, 3 \tag{6}$$

And η_i are design parameters. After few calculations, control law would be computed as:

$$U = \hat{B}^{-1}(\ddot{X}_d - \lambda\dot{\tilde{X}} - \hat{F} - K \operatorname{sgn}(s)) \ , \ \tilde{X} = X - X_{desired} \tag{7}$$

K can be evaluated as:

$$K = \begin{bmatrix} F_1 + \eta_1 + \sum_{j=1}^{3} D_{1j} \left| \ddot{x}_{1d} - \lambda_1 \dot{\tilde{x}}_1 - \hat{f}_1 \right| \\ F_2 + \eta_2 + \sum_{j=1}^{3} D_{2j} \left| \ddot{x}_{2d} - \lambda_2 \dot{\tilde{x}}_2 - \hat{f}_2 \right| \\ F_3 + \eta_3 + \sum_{j=1}^{3} D_{3j} \left| \ddot{x}_{3d} - \lambda_3 \dot{\tilde{x}}_3 - \hat{f}_3 \right| \end{bmatrix} \tag{8}$$

which $\eta_i > 0$ should satisfy sliding conditions, (6).

B. HOSM controller design for a MIMO system

The concept of HOSM first presented by Emelyanov et al. in 1986 [14]. After that, Levant [15] developed the method for SISO systems, and finally, a step-by-step design for MIMO systems introduced in [9].

State equation is as follows:

$$\dot{x} = f(x) + \sum_{i=1}^{m} g_i(x)u_i \ , \ i = 1, \ldots, m \tag{9}$$

which x is an n-dimensional state vector, and u is an m-dimensional control input vector. One can define $\sigma_i(\mathrm{x})$ as sliding variables, which actually are the system outputs, and $g_i(\mathrm{x})$ are smooth uncertain functions. The relative degree vector for PUMA 560 is $r = \begin{bmatrix} 2 & 2 & 2 \end{bmatrix}^T$. One can define B as:

$$B(x) = \begin{bmatrix} L_{g1}L_f\sigma_1 & L_{g2}L_f\sigma_1 & L_{g3}L_f\sigma_1 \\ L_{g1}L_f\sigma_2 & L_{g2}L_f\sigma_2 & L_{g3}L_f\sigma_2 \\ L_{g1}L_f\sigma_3 & L_{g2}L_f\sigma_3 & L_{g3}L_f\sigma_3 \end{bmatrix} \tag{10}$$

which is a non-singular matrix and also:

$$L_{gj}L_f^k\sigma_i(x) = 0 \ , \ 1 \le i, j \le 3 \ , \ 0 \le k < r_{i-1} \tag{11}$$

One may define:

$$A(x) = \begin{bmatrix} L_f^2\sigma_1 & L_f^2\sigma_2 & L_f^2\sigma_3 \end{bmatrix}^T \tag{12}$$

In this point, one can evaluate the r-*th* derivative for each sub-system:

$$\begin{bmatrix} \sigma_1^{(r_1)} & \ldots & \sigma_m^{(r_m)} \end{bmatrix}^T = A(x) + B(x)u \tag{13}$$

To characterize uncertainties, one can write:

$$\begin{cases} A(x) = \bar{A}(x) + \Delta_A(x) \\ B(x) = \bar{B}(x) + \Delta_B(x) \end{cases} \tag{14}$$

which $\Delta_A(x), \Delta_B(x)$ are representing uncertain parts, and $\bar{A}(x), \bar{B}(x)$ are referring to nominal parts. We should also have two criteria:

$$\begin{cases} \left\| \Delta_A(x) - \Delta_B(x)\overline{B}^{-1}(x)\overline{A}(x) \right\| \le \rho(x) \\ \left\| \Delta_B(x)\overline{B}^{-1}(x) \right\| \le 1 - \alpha \end{cases} \tag{15}$$

And finally, the control law may compute as:

$$u = \overline{B}^{-1}(\omega - \overline{A}) \tag{16}$$

Higher Order Sliding Mode control would be as follows:

$$\begin{cases} \begin{cases} \dot{z}_{1,i} = z_{2,i} \\ \vdots \\ \dot{z}_{r_i-1,i} = z_{r_i,i} \end{cases} \\ \begin{bmatrix} \dot{z}_{r_1,1} & \dot{z}_{r_2,2} & \cdots & \dot{z}_{r_m,m} \end{bmatrix}^T = \\ \left[I_m + \Delta_B(x)\overline{B}^{-1} \right]\omega - \Delta_B(x)\overline{B}^{-1}\overline{A}(x) + \Delta_A(x) \end{cases} \tag{17}$$

$$z_{j,i} = \sigma_i^{(j-1)}, \; z_i = \begin{bmatrix} z_{1,i} & z_{2,i} & \cdots & z_{r_i,i} \end{bmatrix}^T$$

$$z = \begin{bmatrix} z_1^T & z_2^T & \cdots & z_m^T \end{bmatrix}^T, i = 1,2,3, \; j = 1,2$$

Defining ω as:

$$\begin{cases} \omega(z) = \omega_{nom}(z) + \omega_{disc}(z, z_{aux}) \\ \dot{z}_{aux} = -\omega_{nom}(z) \end{cases} \tag{18}$$

For ω_{nom} ,we have:

$$\omega_{nom,i}(z_i) = -\left\{ k_{1,i}\operatorname{sgn}(z_{1,i})\left|z_{1,i}\right|^{v_{1,i}} + k_{2,i}\operatorname{sgn}(z_{2,i})\left|z_{2,i}\right|^{v_{2,i}} \right\} \\ i = 1,2,3 \tag{19}$$

such that $p^2 + k_{2,i}p + k_{1,i}$ is Hurwitz, and:

$$v_{1,i} = \frac{v_{2,i}}{2 - v_{2,i}}, \; 1 > v_{2,i} > 0, \\ i = 1,2,3 \tag{20}$$

For the discontinuous part of ω, ω_{disc} we have:

$$\omega_{disc} = -G(z)\operatorname{sign}(S) \tag{21}$$

which S and $G(z)$ may compute as:

$$S = \begin{bmatrix} z_{r_1,1} & z_{r_2,2} & z_{r_3,3} \end{bmatrix} \tag{22}$$

$$G(z) \ge \frac{(1-\alpha)\left\|\omega_{nom}(z)\right\| + \rho(x) + \gamma}{\alpha}, \; \gamma > 0 \tag{23}$$

In this point, one can calculate ω , and then, according to (16), the control signal would be available.

IV. Particle Swarm Optimization

Particle Swarm Optimization (PSO) in an evolutionary algorithm first presented by Kennedy and Eberhart [16]. PSO was inspired from flocks of birds or schools of fish looking for food. The main concept behind PSO is that birds (solutions) are flown through the search space to find feasible and optimum answers. The global best value of each particle ($p_{best,i}$), and the global best value of all particles (g_{best}) be stored then to find the best answer. The basic steps of PSO are as follows [11]:

1) Define a swarm of particles in the search space with random positions.
2) Define a fitness function for comparing the obtained solutions.
3) Evaluate the fitness function for current solutions and update the particle's best value ($p_{best,i}$).
4) Draw a comparison among all best solutions ($p_{best,i}$) to find the best solution so far, which is a solution with minimum fitness function value, and update the global best value (g_{best}).
5) Calculate the next displacement of particles according to their current speed, their best value ($p_{best,i}$), and the global best value (g_{best}). A possible equation for the next displacement of each particle would be as:
6) $$\begin{cases} v_d^{(i+1)} \leftarrow c_1 * v_d^{(i)} + r_1 * c_2(p_d^{(i)} - x_d^{(i)}) + \\ r_2 * c_3(g_d - x_d^{(i)}) \\ v_d^{(i+1)} \leftarrow x_d^{(i)} + v_d^{(i)} \end{cases} \tag{24}$$
   which d is the dimension of the search space, i is the number of current iteration, r_1, r_2 are random numbers between (0,1), c_1, c_2 and c_3 are constants, and $x_d^{(i)}$ is the position of the particle in the *d-th* dimension, and finally, $v_d^{(i)}$ is the speed of the particle for the next displacement.
7) If the ending criteria have been met, return the best values. Otherwise, jump to number (3).

In this paper, a PSO with swarm of 20 particles has been used to optimize the control parameters. The number of iterations is 100 and number of informants would be 3 for each particle. The confinement rules have been applied in order to prevent the particle from leaving the search space. The performance index is the same as fitness function which is the total of squared errors of all joint variables. The optimal parameters for both methods are presented in Tables I and II.

V. Simulation Results

Simulation results for both techniques are presented using MATLAB m-files.

A. *Standard SMC results*

In order to mitigate the Chattering phenomenon, a boundary layer with magnitude of 0.03 [4] has been established. All uncertainty functions in the simulations are of Sine nature. The desired path which showed in Fig. 1 is as follows:

$$\begin{cases} p_x = 0.3 + 0.2\sin(\pi t/5) \\ p_y = 0.3 + 0.2\cos(\pi t/5), \; t > 0 \\ p_z = 0.3 + 0.2\cos(\pi t/5) \end{cases} \tag{25}$$

Nominal model of PUMA 560 has been optimized using PSO. The control parameters for both optimized and non-optimized controllers are gathered in Table I, and control effort and error of position control depicted in Fig. 2 and Fig. 3.

B. HOSM controller results

The biggest advantage of HOSM is its chattering-free nature. Design parameters of both optimized and non-optimized are presented in Table I, and control effort and error of position control showed in Fig. 4 and Fig. 5.

As mentioned before, in order to have a better comparison, both methods are optimized using Particle Swarm Optimization. In this context, optimization means running the nominal model of PUMA 560 repeatedly in order to minimize the total energy of control signals, which showed by $\sum \tau^2$ in Table III. The terminating criteria are either reaching 100 iterations or detecting a steady state error for the position control less than 1 mm, which both are sound criteria in simulation results.

For further examination, a disturbance with a magnitude of 20% of the desired values and a duration of 5 seconds applied to the state equations in order to study of the robustness. The robustness of both systems are perfect and both methods can attenuate the disturbance quickly, but HOSM is faster in this regard.

The response speed is another concern to investigate. The response speed of HOSM is 3.04 seconds, which is a very rapid response. Response speed of SMC is about 3.9 seconds which reveals the superiority of HOSM in this matter. Table III represents a better interpretation of both methods in this discussion. Both accuracy and settling time have been improved, but control effort index of HOSM is almost twice as big as that of standard SMC. This is one of the reasons why the settling time (response speed) and accuracy of HOSM is better than SMC. Another possible reason is the nature of the control design of HOSM which is also able to eliminate the chattering phenomenon. In high percentage of cases which only steady state errors matter, standard SMC would be economically more efficient over HOSM method. Save for that merit, standard SMC has been improved in its novel version, HOSM.

TABLE I. DESIGN PARAMETERS OF STANDARD SMC

Param. / Method	λ_1	λ_2	λ_3	η_1	η_2	η_3
Optimal SMC	2.06	1.325	1.9148	0.080	0.0572	0.0364
Non-optimal SMC	5	5	5	0.001	0.001	0.001

TABLE II. DESIGN PARAMETERS OF HOSM

Parameter / Method	Optimal HOSM	Non-optimal HOSM
K_{11}	12.2000	12
K_{12}	4.5000	12
K_{13}	11.7160	12
K_{21}	9.5568	7
K_{22}	3.0525	7
K_{23}	11.1818	7
V_1	0.6148	0.9
V_2	0.4702	0.9
V_3	0.4207	0.9

TABLE III. COMPARISON OF PERFORMANCE INDICES

Performance Index / Method	$\sum \tau^2$	$\sum e^2(q)$	Settling Time (Sec)
Optimal HOSM	6.7063e+005	243.8369	3.04
Optimal standard SMC	2.8033e+005	592.5962	3.87

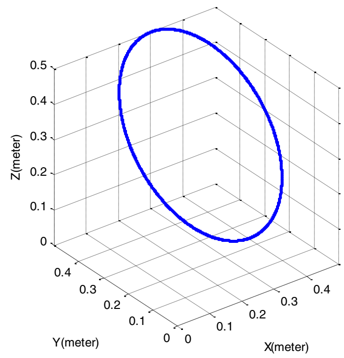
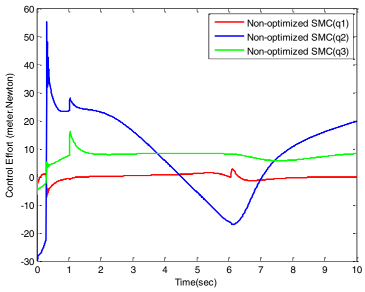
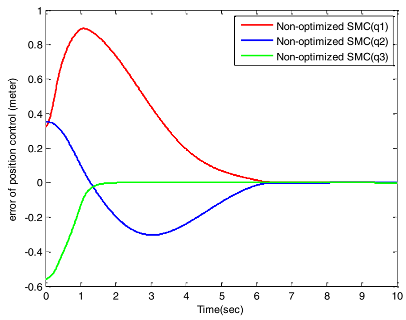


Figure 1. The desired path

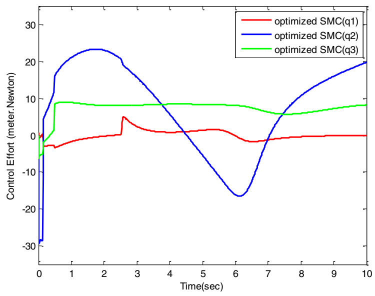


(a) Control effort

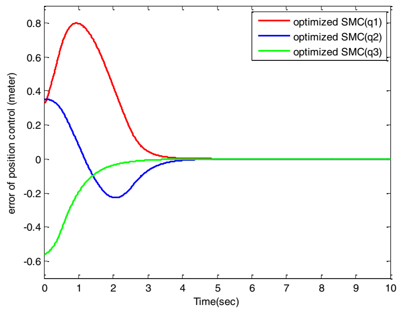


(b) Error of position control

Figure 2. Non-optimized standard SMC: (a) control effort (b) error of position control

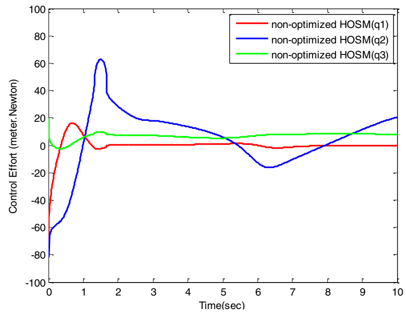


(a) Control effort

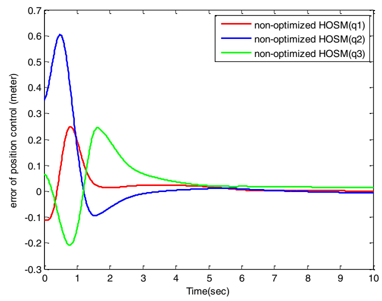


(b) Error of position control

Figure 3.Optimized standard SMC: (a) control effort (b) error of position control

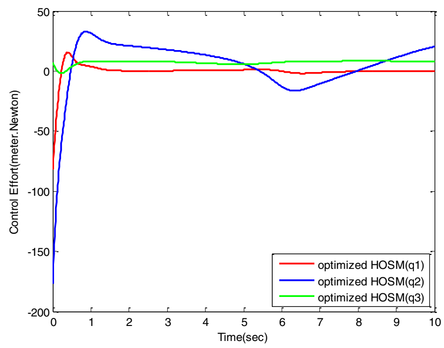


(a) Control Effort

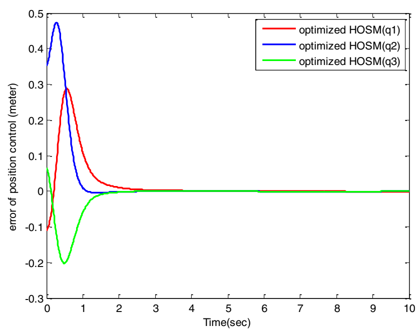


(b) Error of position control

Figure 4. Non-optimized HOSM: (a) control effort (b) error of position control



(a) Control effort



(b) Error of position control

Figure 5. Optimized HOSM: (a) control effort (b) error of position control

VI. CONCLUSION

A comparison has been drawn between two different types of controllers that designed based on sliding mode concept. To draw a distinction between HOSM and standard SMC, the simulation results for PUMA 560 have been presented for both methods, and in order to have a better comparison, both control strategies are optimized using Particle Swarm Optimization. From an energy-saving point of view, the standard SMC needs smaller amount of control effort in comparison to HOSM control scheme. The HOSM technique, on the other hand, shows better accuracy and settling time. Moreover, the control signal would be chattering-free. Although HOSM has more advantages over standard SMC, the latter is economically efficient in comparison to the former.

REFERENCES

[1] V.I. Utkin , "Sliding modes and their application in variable structure systems", *Mir Publishers*, Moscow, 1978.

[2] H. Lee, V. I. Uktin, & A. Malinin, “Chattering reduction using multiphase sliding mode control” *International Journal of Control* , 82 (9), 2009, 1720-1737.

[3] V.I. Utkin, *Sliding Modes In Control And Optimization*, Springer Verlag, Berlin, 1992.

[4] J.J. Slotine, W. Li , *Applied Nonlinear control*, Prentice Hall, 1991, pp. 303-306,.

[5] S. V. Emel'yanov, S. V. Korovin and L. V. Levantovsky, “Higher order sliding modes in control system”, *Differential Equations*, 29(11), 1993, pp. 1627-1647,.

[6] L. Fridman and A. Levant, “Higher order sliding modes”, In W. Perruquetti and J. P. Barbot (Eds), *Sliding mode control in Engineering*, Marcel Dekker, New York, 2002, pp. 53-101,.

[7] G. Bartolini, A. Pisano, E. Punta and E. Usai, “A survey of applications of second order sliding mode control to mechanical systems”, *Int. J. of Control*, 76(9), 2003, pp. 875-892,.

[8] T. Floquet, J. P. Barbot and W. Perruquetti, “Higher order sliding mode stabilization for a class of nonholonomic perturbed systems”, *Automatica*, 39(6), 2003, pp. 1077-1083.

[9] Michael Defoort, Thierry Floquet, Annemarie M. Kokosy, Wilfrid Perruquetti, “A novel higher order sliding mode control scheme”, *Systems & Control Letters* 58(2) 2009, pp. 102-108.

[10] Farzin Piltan , M.H. Yarmahmoudi, M. Shamsodini, E.Mazlomian, A.Hosainpour. “PUMA-560 Robot Manipulator Position Computed Torque Control Methods Using MATLAB/SIMULINK and Their Integration in Graduate Nonlinear Control and MATLAB Courses”, *International Journal of Robotics and Automation*, 2012, 3(3):

[11] M. Clerc, *Particle Swarm Optimization*, John Wiley & Sons Inc (US), 2006.

[12] Mark W.Spong , Seth Hutchinson, and Mathukumalli Vidyasagar. “Robot modeling and control*”*. New York: *John Wiley & Sons*, 2006.

[13] CS George Lee and M. Ziegler, “Geometric approach in solving inverse kinematics of PUMA robots”. *Aerospace and Electronic Systems, IEEE Transactions on,* 1984, 6: 695-706. .

[14] S.V. Emelyanov, S.K. Korovin and L.V. Levantovsky, “Higher order sliding modes in the binary control systems”, *Soviet Physics*, Doklady, 1986, pp. 291-293.

[15] Arie Levant. "Sliding order and sliding accuracy in sliding mode control." *International journal of control* 58.6 (1993) pp. 1247-1263.

[16] J. Kennedy, R. Eberhart, et al., “Particle swarm optimization”, in *Proceedings of IEEE international conference on neural networks*, vol. 4, 1995, pp. 1942–1948.