International Journal of Engineering Innovation & Research
Volume 3, Issue 1, ISSN: 2277 – 5668

Attitude Control of a Rigid Satellite Using Intelligent Change of Angular Momentum and Comparison with Classic Results Regarding to LQR

K. Shojaee G.
Ph.D. Student, Computer Engineering,
Ferdowsi University of Mashhad, Iran
Email: shojaee@stu.um.ac.ir

H. Rajabi Mashhadi
Professor of Electrical Engineering,
Ferdowsi University of Mashhad, Iran
Email: h-mashhadi@um.ac.ir

***Abstract* – One of the maneuvers that a satellite should be able to do is the change of the satellite attitude or its body coordinates according to a reference one. This maneuver is possible by using the proper control torques and mechanism reaction wheels. This system is a multivariable and nonlinear system and the coupled equations governing the rotational motion of the satellite complicate the problem. In this paper, the optimal control policy and the potential advantages and disadvantages of both methods are compared using two different approaches. In this study, using Mushy State Simulated Annealing (MSSA) algorithm as an intelligent optimization method, a new concept is presented to face the Dynamic Programming approach which is a low speed and massive method from computational view. On the other hand, the problem is solved based on analytical methods discussed in optimized control issues and by integrating the Receding Horizon Control (RHC) and Linear Quadratic Regulator (LQR) concepts. Then the results are compared.**

***Keywords* – Linear Quadratic Regulator, Mushy State Simulated Annealing, Attitude Control, Satellite, optimization.**

I. INTRODUCTION

The current generation of satellites and spacecrafts requires being equipped with the attitude control systems to be able to perform duties such as attitude maneuvers and tracking. This mission is based on a sequence of orbital maneuvers around each control axis which creates the desired rotation. Optimal control approach is highly regarded in recent researches for attitude maneuver of the satellites. These investigations are mainly with the aim of optimizing the time and fuel consumptions. Some of the applications are given in refs. [1-6].

In control of a spacecraft with minimal energy consumption, Seywald et al. [7] investigated a fuel-optimal solution for the reorientation of a symmetric rigid spacecraft with independent three-axes controls. Vadali and Junkins [4] studied a large-angle reorientation of rigid asymmetric spacecraft with multiple reaction wheels in which they used an integral of weighted quadratic functions as the cost function. In practice, the limitations of control input data should be considered. In general, the problems of optimal control with limited control for symmetrical and asymmetric satellites or spacecraft are solved using the Pontryagin's principle [8]. To find the control commands in these problems with optimal control approach, many articles use the minimization of a cost function in a fixed period [4],[9]-[14]. In this paper, the maneuver period is fixed 45 seconds.

In this study, determination of the control policies is discussed using two different approaches. At first, using the analytical methods in optimal control with the concept of RHC, the problem starts. In each step, the equations are linearized around the current situation and the proper control command for the considered period of time is produced using LQR method and RHC idea. This method results in the closed loop control which has a good resistance against disturbances and is applicable to a nonlinear system. In continue, a new version of simulated annealing (SA) method is used in order to intelligently determine the control policies in comparison with the dynamic programming database which significantly reduces the calculations size.

II. LQR METHOD AND RHC ALGORITHM

A. LQR Method

Linear Quadratic Regulator (LQR) method is a well-known and basic issue in optimal control theory. For decades, many researchers have tried to develop the principles of LQR control [15]-[17]. In LQR method, a cost function is produced with definition of a standard performance measure which is shown in Eqs. (1) and (2). Optimal control policies are generated in such a way that will minimize the cost function. The cost function can be defined as continuous or discrete:

$$J=\frac{1}{2}X^T(t_f)HX(t_f)+\int_{t_0}^{t_f}\frac{1}{2}[X^T(t)Q(t)X(t)+U^T(t)R(t)U(t)]dt \quad (1)$$

$$J=\frac{1}{2}X^T(N)HX(N)+\frac{1}{2}\sum_{k=0}^{N-1}[X^T(k)Q(k)X(k)+U^T(k)R(k)U(k)] \quad (2)$$

Where, H, Q and R are weight matrices that their values depend on the relative importance of system states control. It is required in the algorithm that some or all system state variables converge to zero. The entries of weight matrices H, Q and R values are defined and designed proportion to the importance of each state variable in having a zero value. Thus the state variable that is more important to become zero, its corresponding entry in weight matrices is larger and its distance from the zero costs more. Therefore, the control commands are necessarily produced in such a way that the desired mode produces the lowest possible cost. In this method, the control commands are generated based on the idea of dynamic programming from the end to the beginning like a sequence of control commands. So the control policies are generated offline. By applying this control series to the system, it is expected that the states move and change toward the desired values.

B.RHC algorithm

The main concept of the RHC for regulation problems is briefly the solution of an optimization problem for a limited duration in future and generation of a control policy for change in the current system state in which only the control commands are selected and applied from the generated control series corresponding to the current state. This process will be repeated in each period to confirm that the time horizon is moving forward in each iteration.

Because of on-line calculations of control commands, RHC has many privileges such as high efficiency in tracking, control of uncertainty, solving nonlinear time-dependent systems [18]-[23]. Moreover, RHC is generated based on the current state which causes a closed loop control [21].

III. EQUATIONS & OPTIMIZATION PROCEDURE

In this article, three reaction wheels are assumed for a rigid satellite that each of them is installed and adjusted along one of the main axes of the body coordinates to be able to control the satellite's orientation in space. So the dynamic equations governing this rigid satellite and its angular momentum of the reaction wheels will be as follows:

$$(J_1 - I_1)\dot{\Omega}_1 = (J_2 - J_3)\Omega_2\Omega_3 - h_3\Omega_2 + h_2\Omega_3 - u_1 \quad (3)$$

$$(J_2 - I_2)\dot{\Omega}_2 = (J_3 - J_1)\Omega_3\Omega_1 - h_1\Omega_3 + h_3\Omega_1 - u_2 \quad (4)$$

$$(J_3 - I_3)\dot{\Omega}_3 = (J_1 - J_2)\Omega_1\Omega_2 - h_2\Omega_1 + h_1\Omega_2 - u_3 \quad (5)$$

$$\dot{h}_{1,2,3} = u_{1,2,3} - I_{1,2,3}\dot{\Omega}_{1,2,3} \quad (6)$$

Where, J_1, J_2, J_3 and I_1, I_2, I_3 are the moment of inertia of the satellite and reaction wheels about the main axes of body coordinate system respectively. Ω_1, Ω_2 and Ω_3 are the angular velocities of the satellites body system. The relative angular momentum of the wheels around the specific body axes are shown with $h_i i=1,2,3$andu_1, $u_2,$ u_3 are the wheels input torques.

In order to analyze this system, its state vector is defined as below:

$$\vec{X}^T = \begin{bmatrix}\Omega_1 & \Omega_2 & \Omega_3 & \varphi & \theta & \psi & h_1 & h_2 & h_3\end{bmatrix} \quad (7)$$

$$\dot{\Omega}_1 = \frac{1}{(J_1 - I_1)}\left[(J_2 - J_3)\Omega_2\Omega_3 - h_3\Omega_2 + h_2\Omega_3 - u_1\right] \quad (8)$$

$$\dot{\Omega}_2 = \frac{1}{(J_2 - I_2)}\left[(J_3 - J_1)\Omega_3\Omega_1 - h_1\Omega_3 + h_3\Omega_1 - u_2\right] \quad (9)$$

$$\dot{\Omega}_3 = \frac{1}{(J_3 - I_3)}\left[(J_1 - J_2)\Omega_1\Omega_2 - h_2\Omega_1 + h_1\Omega_2 - u_3\right] \quad (10)$$

$$\dot{\varphi} = (\Omega_2 \sin\varphi + \Omega_3 \cos\varphi)\tan\theta + \Omega_1 \quad (11)$$

$$\dot{\theta} = \Omega_2 \cos\varphi - \Omega_3 \sin\varphi \quad (12)$$

$$\dot{\psi} = (\Omega_2 \sin\varphi + \Omega_3 \cos\varphi)\sec\theta \quad (13)$$

$$\dot{h}_1 = u_1 - \frac{I_1}{(J_1 - I_1)}\left[(J_2 - J_3)\Omega_2\Omega_3 - h_3\Omega_2 + h_2\Omega_3 - u_1\right] \quad (14)$$

$$\dot{h}_2 = u_2 - \frac{I_2}{(J_2 - I_2)}\left[(J_3 - J_1)\Omega_3\Omega_1 - h_1\Omega_3 + h_3\Omega_1 - u_2\right] \quad (15)$$

$$\dot{h}_3 = u_3 - \frac{I_3}{(J_3 - I_3)}\left[(J_1 - J_2)\Omega_1\Omega_2 - h_2\Omega_1 + h_1\Omega_2 - u_3\right] \quad (16)$$

These equations are non-linear and coupled. For linearization, it is needed for the nonlinear equations to be linearized around the operating points. While in this problem, merely the initial and final state of the system is obvious.

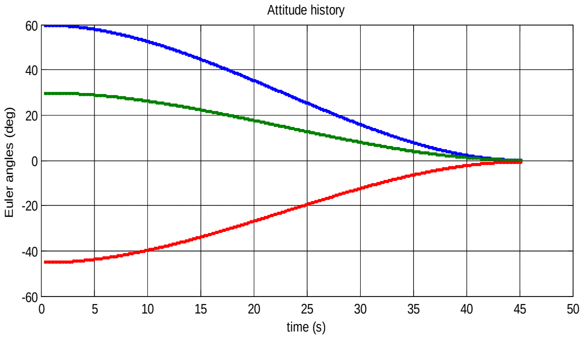

Fig.1. Euler angle vs. time

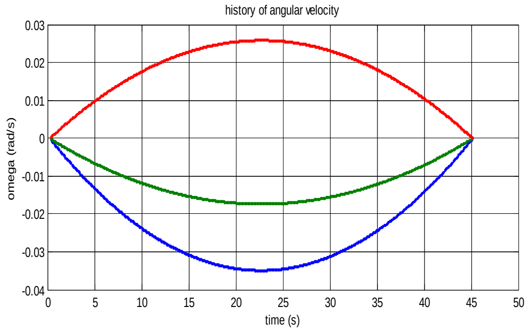

Fig.2. Angular velocity vs. time

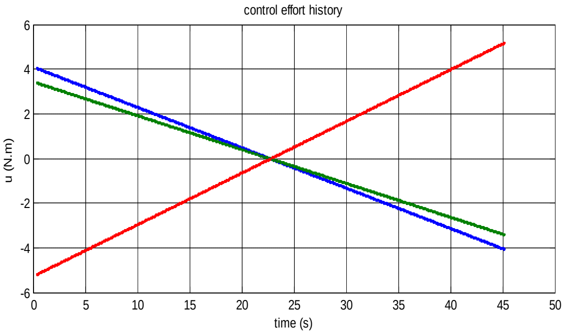

Fig.3. Control policy vs. time

Therefore, a solution is needed to produce the operating points and linearize the equations around them. The method which is used in this paper for creating the operating points is that with discretizing and linearizing the equations around the initial conditions, control commands series are obtained based on LQR method. In order to use the LQR method, it is needed to form the cost function based on the standard form of LQR problems. Due to the discrete approach, the costs function is defined as Eq. (2) in which *H* and *Q(k)* are real, symmetric, positive and semi-definite *n* by *n* matrices. *R(k)* is a real, symmetric, positive and definite *m* by *m* matrix. *n* and *m* are the number of system state variables and the number of control actions, respectively. *N* is a real positive number.

For the above cost function, the optimum control policy between steps *N*-*K* to *N* is obtained by:

$$u^*(N-K) = -[R(N-K)+B^T(N-K)P(K-1)B(N-K)]^{-1}$$
$$\times B^T(N-K)P(K-1)A(N-K)X(N-K) = F(N-K)X(N-K) \tag{17}$$

And the minimum cost between steps N-K to N will be:

$$J^*_{N-K,N}(X(N-K)) = \frac{1}{2}X^T(N-K)\{[A(N-K)$$
$$+B(N-K)F(N-K)]^T$$
$$\times P(K-1)[A(N-K)+B(N-K)F(N-K)]$$
$$+F^T(N-K)R(N-K)F(N-K) + Q(N-K)\}X(N-K)$$
$$= \frac{1}{2}X^T(N-K)P(K)X(N-K) \tag{18}$$

To calculate the feedback:

$$F(N-K) = -[R(N-K)+B^T(N-K)P(K-1)B(N-K)]^{-1}$$
$$\times B^T(N-K)P(K-1)A(N-K) \tag{19}$$

In which matrix P will be obtained by:

$$P(K) = [A(N-K)+B(N-K)F(N-K)]^T P(K-1)$$
$$\times[A(N-K)+B(N-K)F(N-K)]$$
$$+F^T(N-K)R(N-K)F(N-K)+Q(N-K) \tag{20}$$

Where *P(0)=H* and *F(N-1)* is computable using *P(0)* and Eq. (27). The optimum control policy$U^*(N-1)$ which means the control policy that moves the state to the final state is obtained by multiplication of $F(N-1)$ and$X(N-1)$. So, as it is observed, the last control policy is computable at first. *P(1)* is obtained by using $F(N-1)$ and Eq. (28) and then $F(N-2)$and consequently P(2). This process will continue until primary control policy is produced ultimately [24]. Since the system is linearized piece to piece, according to RHC approach, only the initial control policy is important in order to change the system from the current state to the next one and linearize the system around the new state.

Therefore, based on RHC idea for the current system state, the corresponding control command is selected among the generated control series. By applying this control command to the nonlinear system, the new state is produced which can be used as the next operating point. This is continued through a loop in the program until achievement to the final state of zero.

Fig.1 shows the attitude history with the Euler angle changes of the satellite in a 45 second period. As it is seen, the angles are regulated from their initial states to zero in such a way that the cost function becomes minimized.

Fig.2 shows angular velocity changes versus time. Angular velocities increase from their initial state which is zero and will be regulated again toward zero. Fig. 3 shows the control policy in 45 seconds.So a rigid satellite is rotated optimally from its initial state to the favorite one using these control commands.

IV. MUSHY STATE SIMULATED ANNEALING ALGORITHM

Various methods are possible for optimization of cost function. In order to use the dynamic programming approach, it is possible to quantize the system in the problem defined space considering the restrictions governing the control commands. Then the cost function is calculated in different states with various control actions by generating a database and finally the optimum control policy is found. But because of the large number of states and their wide range, the database will be so massive and generation of such a database would be virtually impossible by means of available computing tools. To deal with this problem, intelligent search methods can be used to minimize cost functions among possible states and control commands.

In this paper, in addition to analytically prove by linearization of the equations around the operating points, an intelligent search is also applied by Mushy State Simulated Annealing method to obtain the optimal control policy. Since linearization in order to solve non-linear equations always has some error, soft computing approach to achieve optimal control policy will be effective. This method has a high capability in convergence to absolute minimum among other intelligent methods [25].

In this section, mushy state simulated annealing method is introduced which is used for the first time in his paper for non-continuous problems because of its higher convergence speed in comparison with simulated annealing method.

Mushy state simulated annealing is a modified algorithm of simulated annealing method which is one of the of Monte carlo approaches in optimization of multi-variable equations. Its name refers to cooling procedure of a molten metal near to the freezing point. These revolutionary algorithmsare first used in 2009 in solving of a traveling salesman problem which is in NP-Complete problem category [25]. In this process, the aim is to approach the equilibrium state if crystalline structure with minimum energy. In a cooling process, the metal is heated to its melting point and then slowly cooled in such a way that the system is always in a thermodynamically equilibrium state [26],[27].

The main Metropolis scheme in determination of mode of heat and power of the primary thermodynamic systems is that if the energy change is negative, the new structure (energy and temperature) will be accepted otherwise the acceptance will be subject to a Boltzmann probability function $exp(-\Delta E/\kappa T)$. k is the Boltzmann constant with a positive value [28]. This process is repeated until the energy is minimized and the system reaches the equilibrium state. SA algorithm is controlled by cooling schedule parameters. It can give a nearly optimum response in a reasonable time. This method is a stochastic one which is suitable for mixed discrete problem and complicated nonlinearity problems. The most important part of SA algorithm is determination of the cooling schedule which is summarized in the following steps:

1. Determine the initial temperature
2- Determine the final temperature (freezing point)
3- Determine the cooling schedule
4- Determine the Markov chain length

In different optimization papers, the basic concepts of scheduling are determined quantitatively and do not follow a specific discipline. In fact, this issue problem dependent and the researchers prefer to optimize their algorithm by changing the above concepts. In this paper, because of the wide search space and the number of the goal function variables, a new approach of SA simulating the mushy sate is used to decrease the total search time. In SA process, the control parameter is temperature. High temperature shows the possibility of great changes in cost function and movement in search space with long steps or movement on the positive slope of goal function is possible in high temperatures. In a mushy state simulated annealing algorithm, the initial conditions of the problem are defined in a way that simulates the state at a temperature near to the mushy state temperature. The SA parameters need to be redefined for MSSA which are discussed in continue.

A.Definition of initial temperature

Cooling process which starts from a high temperature is a time-consuming procedure. In this paper, it is proposed to start the cooling process from a lower temperature with lower energy. This condition can be the mushy state instead of the molten state. In this state the portion of accepted states are less than 10% of the whole tested states at the beginning of the algorithm while in melting point temperature usually 70% to 97% of the total states are accepted [29]-[31]. This temperature is like the freezing state but with a proper definition of initial energy level is can simulate the mushy state.

B.Definition of final temperature

There are several ways to define the freezing conditions (convergence conditions) in SA algorithm. Pao [29] used a value to define final temperature in which the algorithm stops. Thompson [30] obtained the freezing temperature like the initial temperature from the acceptance equation for 0.01. Chen [31] defined the final temperature so that the acceptance probability becomes 10^{-10}. In the MSSA, a condition in which the minimum and maximum accepted values are the same in one Markov chain length is known as the final temperature or convergence condition of the algorithm. it is shown in ref. [29] that it has more efficiency in determination of convergence conditions.

C.Cooling schedule

Table cooling schedule is different in various algorithms and depends on the functions type and its nonlinearity behavior and their changes. Aarts [32] proved the convergence of SA algorithm in case of temperature reduction based on the following pattern.

$$T_{k+1} = T_0/(1+log(k)) \qquad (21)$$

In which $T_0$is the initial temperature and k the outer iteration cycle in which the temperature changes. But to increase the speed, in many researches they have used a cooling schedule $T_{k+1}=\alpha\times T_k$or $T_{k+1} = T_k/(1+log(k))$. Each pattern can show an optimal behavior for a specific application. So in this paper, in order to gain the best accuracy in the least time, the cooling schedule in intermediate temperature is a little bit slower than usual but the initial temperature has also been decreased. This approach significantly decreases the convergence time. The cooling schedule in the optimization algorithm is considered as Eq. (22).

$$T_{k+1} = T_k/(1+log(k^{0.5})) \qquad (22)$$

D. Markov chain length

In this algorithm, by using the memory and the cooling schedule, the Markov chain at which the algorithm repeats the search process at a constant temperature is considered equal to 500.

E. Initial energy level

In this problem, in order to reach the initial energy level required for simulation of the mushy state, it is possible to guess the change trend of the optimal control commands as the simplest form; i.e. linear.to generate this linear trend, that is enough to guess two points of the line to be able to find the equation of straight line. To generate the first point, since the initial system state is specific, we can linearize the nonlinear system equations around this state. Then produce a control series backward using LQR method. The last member of this series is the first control command which can be the required first point.

To change the angle of a stationary satellite from an initial state to the final zero one, a torque is needed so that the satellite starts a gradual rotation. Since the satellite should be in a stationary state at the end of its mission, it is obvious that a negative acceleration should be applied during the process to stop it. So it is assumed that the satellite changes the rotation direction at the middle of mission and the optimum control commands are zero at that time. This way the second point is obtained with which the line equation can be calculated and used in MSSA.

F.Dynamic memory in SA

A new concept of dynamic memory is used in this paper to improve the performance of MSSA. A capability of SA algorithm is to intelligently search toward the positive gradient changes which prevents the algorithm from becoming stuck in local minima. In fact, the “uphill” capability not only increases the efficiency of the algorithm in search space in comparison with other methods, but also highly increases the convergence ratio in case if integration with dynamic memory. By keeping the memory in the search algorithm, it will be possible to keep the best result during the process and search again at temperatures nearer to the freezing point based on that. This will reduce the probability of passing the suitable range without convergence to the optimum response. Too high temperature in SA algorithm highly increases the run time without considerable effect on the convergence accuracy. Inappropriate Markov chain length and cooling schedule cause to entrap in local minima as well as increase the search time. Freezing temperature has the same effect. So there is a correlation among all of the SA parameters. The dynamic memory stabilizes the algorithm against the effects of other parameters and increases the MSSA convergence speed. It saves the best response in a buffer and applies it to the algorithm when the Metropolis acceptance condition is not achieved in a tenth of Markov chain length, so that the algorithm continues searching.

Table1 shows the MSSA steps.

V. DISCUSSION AND COMPARISON

Two analytical methods are compared for a satellite; linearization method and intelligent search by MSSA method. The comparison of results is made by linearization in the same number of points and dividing the cost function by the number of linearized points. To analyze the increase of accuracy and its effects in analytical calculations with increasing the number of linearized points, the results are calculated with defining the operating points from 100 to 1500 points. Table 2 shows the results obtained by LQR and the average of MSSA results for 10 runs.

δ criteria that shows the improvement percentage of MSSA algorithm is defined as Eq. (23).

$$\delta = \left(\frac{f(LQR)-f(MSSA)}{f(MSSA)}\right) * 100 \qquad (23)$$

f is the cost function that will be calculated for the control command series by Eq. (2). *f(LQR)* is the cost function of LQR results and *f(MSSA)* the cost function of MSSA results.

As shown in Table 2, this method gives better control commands with less cost in comparison with LQR method when there are not many divisions in discretizing process (larger time periods). This is one of the privileges of this method. By increasing the number of divisions and decreasing the time period, the LQR method's error is decreased and the responses of both methods converge.

Table I: MSSA algorithm steps

1) Read initial values for mushy state: initial values for input variables vector (P0) at the mushy state energy level, initial temperature for mushy state, input amplitude, neighborhood radius for input variables, neighborhood radius reduction coefficient, markov chain length, Minf=∞, MinP=P0
2) Calculate cost function for input variables vector f(P0)
3) Generate a new random variable vector based on the former one and current neighborhood radius (Pk+1)
4) Calculate new cost function for new variable vector f(Pk+1)
5) Check Metropolis condition: IF f(Pk+1)<f(Pk) THEN accept the new vector
6) IF f(Pk+1)>f(Pk) THEN generate a random number (r) and acceptance probability $p(\Delta(f)) = \exp(\frac{-\Delta(f)}{T})$. IF r<p then accept the new vector and make the non-accepted variables counter zero, otherwise reject the new vector and increase the non-accepted variables counter.
7) IF f(Pk+1)<Minf THEN Minf = f(Pk+1) and MinP=Pk+1.
8) IF non-accepted variables counter is larger than 1/10 of Markov chain length THEN replace the current variable vector with the minimum found energy function (MinP).
9) Increase the Markov chain counter
10) IF the Markov chain counter is less than the Markov chain length THEN repeat the algorithm from Step 3, otherwise continue.
11) Check the freezing (convergence) condition and Stop if satisfied.
12) Change the temperature based on cooling schedule, change the neighborhood radius based on the neighborhood radius reduction coefficient, make the Markov chain counter zero, make the non-accepted variables counter zero.

Fig. 4 shows the results of Table 2. It is observed that the increase of the number of control commands causes to decrease the improvement coefficient δ. MSSA algorithm has a better efficiency up to 7% in comparison with LQR method when the control commands are few in the satellite control period.

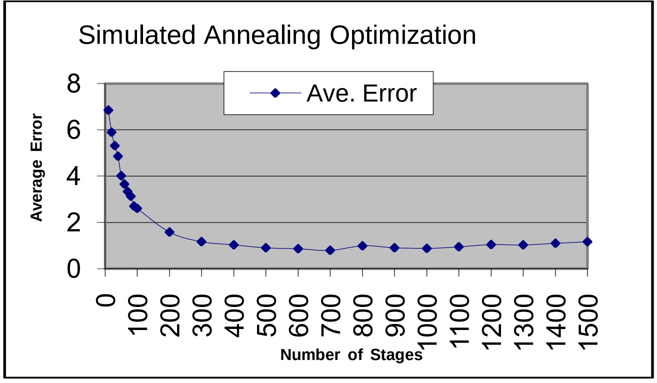

Fig.4. Improvements of the cost function.

Table II: The results of cost function for the control commands obtained by LQR and MSSA methods.

	LQR-RHC		Average of SA Optimization		Ave. of Improve -ment
No. Stages	Cost	Cost/ Nstage	Ave. Cost	Ave. Cost/ Nstage	δ
10	0.515	0.051	0.482	0.048	6.842
20	0.867	0.043	0.819	0.041	5.894
30	1.232	0.041	1.17	0.039	5.306
40	1.599	0.04	1.525	0.038	4.85
50	1.967	0.039	1.891	0.038	4.014
60	2.335	0.039	2.253	0.038	3.659
70	2.703	0.039	2.616	0.037	3.329
80	3.071	0.038	2.978	0.037	3.131
90	3.439	0.038	3.349	0.037	2.702
100	3.807	0.038	3.71	0.037	2.603
200	7.468	0.037	7.353	0.037	1.573
300	11.101	0.037	10.974	0.037	1.163
500	18.282	0.037	18.12	0.036	0.895
600	21.831	0.036	21.647	0.036	0.854
700	25.353	0.036	25.155	0.036	0.79
800	28.849	0.036	28.566	0.036	0.989
900	32.317	0.036	32.028	0.036	0.902
1000	35.76	0.036	35.448	0.035	0.88
1100	39.176	0.036	38.811	0.035	0.943
1200	42.567	0.035	42.127	0.035	1.046
1300	45.932	0.035	45.466	0.035	1.026
1400	49.272	0.035	48.738	0.035	1.096
1500	52.587	0.035	51.982	0.035	1.163

VI. CONCLUSION

In this paper, a cost function was defined by Eq. (2) for attitude control of a satellite to find the appropriate control commands for its rotation. The control commands and system states have to be generated in such a way to produce the minimum cost function. In this paper, the optimal control commands for attitude control of a satellite are generated by two different methods. First they are obtained off-line and on-line for a multi-variable nonlinear system by integrating LQR and RHC method. In the second approach, the control commands are produced using MSSA method so that the minimum cost function is obtained. In larger time periods, this method gives lower costs and more optimal commands. By decreasing the time periods in order to discretize the problem, the costs obtained by both methods are converged which approves the production of the optimum response with both methods.

REFERENCES

[1] M.V. Dixon, T. Edelbaum, J. E Potter, W.E. Vandervelde, Fuel optimal reorientation of axis symmetric spacecraft, *J. Spacecr.* 7 (1970) 1345–1351.

[2] J.L. Junkins, J.D. Turner, Optimal continuous torque attitude maneuvers, *J. Guidance Control Dyn.* 3 (3) (1980) 210–217.

[3] S.R. Vadali, L.G. Kraige, J.L. Junkins, New results on the optimal spacecraft attitude maneuver problem, *J. Guidance Control Dyn.* 7 (3) (1984) 378–380.

[4] S.R. Vadali, J.L. Junkins, Optimal open-loop and stable feedback control of rigid spacecraft attitude maneuvers,*J. Astronaut. Sci.* 32 (1984) 105–122.

[5] F. Li, P.M. Bainum, Numerical approach for solving rigid spacecraft minimum time attitude maneuvers, *J. Guidance Control Dyn.* 13 (1) (1994) 38–45.

[6] S.L. Scrivener, R.C. Thomson, Survey of time-optimal attitude maneuvers, *J. Guidance Control Dyn.* 17 (2) (1994) 225–233.

[7] H. Seywald, R.R. Kumar, S.S. Deshpande, M.L. Heck, Minimum fuel spacecraft reorientation,*J. Guidance Control Dyn.*17 (1) (1994) 21–29.

[8] L.S. Pontryagin, V.G. Boltyanskii, R.V. Gamkrelidze, E.F. Mishchenko, The Mathematical Theory of Optimal Processes, *Gordon and Beach, New York,* 1986.

[9] T.A.W. Dwyer, The control of angular momentum for asymmetric rigid bodies, *IEEE Trans. Autom. Control 3* (1982) 686–688.

[10] Y.Y. Lin, L.G. Kraige, Enhanced techniques for solving the two-point boundary-value problem associated with the optimal attitude control of spacecraft, *J. Astronaut. Sci.* 37 (1989) 1–15.

[11] C.K. Carrington, J.L. Junkins, Optimal nonlinear feedback control for spacecraft attitude maneuvers, *J. Guidance Control Dyn.* 9 (1) (1986) 99–107.

[12] S.R. Vadali, J.L. Junkins, Spacecraft large angle rotational maneuvers with optimal momentum transfer, *J. Astronaut. Sci.* 31 (1983) 217–235.

[13] H. Schaub, J.L. Junkins, New penalty functions and optimal control formulation for spacecraft attitude control problems, *J. Guidance Control Dyn.* 20 (3) (1997) 428–434.

[14] T.A.W. Dwyer, Exact nonlinear control of spacecraft slowing maneuvers with internal maneuver transfer, *J. Guidance Control Dyn.* 9 (2) (1986) 240–247.

[15] B.D.O. Anderson, J.B. Moore, Optimal Control: Linear Quadratic Methods, *Prentice-Hall International, London,* 1989.

[16] A.E. Bryson, Y.C. Ho, Applied Optimal Control: Optimization, Estimation, and Control, *John Wiley & Sons, New York,* 1975.

[17] F.L. Lewis, V.L. Syrmos, Optimal Control, *John Wiley & Sons, New York,* 1995.

[18] M.V. Kothare, V. Balakrishnan, M. Morari, Robust constrained model predictive control using linear matrix inequalities, *Automatica* 32 (10) (1996) 1361–1379.

[19] W.H. Kwon, A.E. Pearson, A modified quadratic cost problem and feedback stabilization of a linear system, *IEEE Trans. Automat. Control* 22 (5) (1977) 838–842.

[20] W.H. Kwon, A.E. Pearson, On feedback stabilization of time-varying discrete linear systems, *IEEE Trans. Automat. Control* 23 (3) (1978) 479–481.

[21] W.H. Kwon, S. Han, Receding Horizon Control: Model Predictive Control for State Models, *Springer-Verlag, London,* 2005.

[22] W.H. Kwon, K.B. Kim, On stabilizing receding horizon controls for linear continuous time-invariant systems, *IEEE Trans. Automat. Control* 45 (7) (2000) 1329–1334.

[23] J. Richalet, A. Rault, J.L. Testud, J. Papon, Model predictive heuristic control: Applications to industrial processes, *Automatica*14 (5) (1978) 413–428.

[24] Donald E.Kirk; *Optimal Control Theory*, 1st Edition

[25] K. Shojaee G., H. Shakouri G. and M.B. Menhaj" A Mushy State Simulated Annealing"Amirkabir International Journal systems, Modeling, Identification, Simulation &Control, Vol . 43, No.2, Fall 2011, pp. 59-70.

[26] S. Kirkpatrick, C. D. Gelatt, Jr., M. P. Vecchi, "Optimization by Simulated Annealing", SCIENCE, Volume 220, Number 4598, pp. 671-680, 13 May 1983.

[27] V. Cerny, “Thermodynamical Approach to the Traveling Salesman Problem: An Efficient Simulation Algorithm,” J. Opt. Theory Appl, vol.45, pp. 41-51, 1985.

[28] N. Metropolics, A. Rosenbluth, M. Rosenbluth, A. Teller, E. Teller, “Equation of State Calculations by Fast Computing Machines,” J. Chem. Phy, vol.21, pp. 1087-1092, 1953.

[29] D. C. W. Pao, S. P. Lam and A. S. Fong, "Parallel implementation of simulated annealing using transaction processing", IEE Proc-Comput. Digit. Tech.. Vol. 146, No. 2, pp. 107-113, March 1999.

[30] Dale R. Thompson, and Griff L. Bilbro, "Sample-Sort Simulated Annealing", IEEE Transactions On Systems, Man, And Cybernetics—Part B: Cybernetics, Vol. 35, No. 3, Pp. 625-632, June 2005.

[31] Hao Chen, Nicholas S. Flann, and Daniel W. Watson, "Parallel Genetic Simulated Annealing: A Massively Parallel SIMD Algorithm", IEEE Transactions On Parallel And Distributed Systems, Vol. 9, No. 2, Pp.126-136, February 1998.

[32] E. Aarts and J. Korst, "Simulated Annealing and Boltzmann Machines: A Stochastic Approach to Combinatorial Optimization and Neural Computing", *New York: Wiley, 1989*.

AUTHOR’S PROFILE

Kambiz Shojaee G.
was born in Mashhad, Iran, in 1983. He received the B.S. degree in Electronic engineering from the Bahonar University of Kerman, Kerman, Iran, in 2003 and the M.S. degree in Electronic engineering from the the Department of Electrical and Computer Engineering of Tehran University in 2005. At now he is a Ph.D. Student in Computer Engineering at Ferdowsi University.

His research interests are in the areas of evolutionary algorithms, soft computing, multiagent systems, complex and intelligent systems and fuzzy logic and control. His researches lead to some Journal and conference papers. He was referee of some national and international conferences and also he is a student member of Center of Excellence on Soft Computing and Intelligent Information Processing at Ferdowsi University of Mashhad.

Habib Rajabi Mashhadi
obtained his B.Sc. and M.Sc. degrees with honor from the Ferdowsi University of Mashhad, Mashhad, Iran, both in electrical engineering and his PhD from the Department of Electrical and Computer Engineering of Tehran University in 2002. He has been a professor in Electrical Engineering Department and Center of Excellence on Soft Computing and Intelligent Information Processing, Ferdowsi University since then His research interests are Power System Operation and Dynamics and Biological Computation.