UDK 629.5 https://doi.org/10.15407/itm2020.04.043

S. KHOROSHYLOV, M. REDKA

RELATIVE CONTROL OF AN UNDERACTUATED SPACECRAFT USING REINFORCEMENT LEARNING

Institute of Technical Mechanics of the National Academy of Sciences of Ukraine and the State Space Agency of Ukraine, 15 Leshko-Popel St., Dnipro 49005, Ukraine; e-mail: skh@ukr.net , mix5236@ukr.net

Метою статті є апроксимація оптимального керування відносним рухом космічних апаратів при неповному складі виконавчих органів з використанням навчання з підкріпленням і дослідження впливу різних чинників на якість такого рішення.

При проведенні досліджень використані методи теоретичної механіки, теорії автоматичного керування, теорії стійкості, методи машинного навчання та комп'ютерного моделювання.

Розглянуто задачу керування відносним рухом космічних апаратів в площині орбіти з використанням тільки керуючих впливів, спрямованих по дотичній до орбіти. Використання такого підходу дозволяє зменшити витрату робочого тіла реактивних виконавчих органів і спростити архітектуру системи керування, проте в ряді випадків використання методів класичної теорії керування не дозволяє отримувати прийнятні результати. У зв'язку з цим досліджена можливість вирішення цього завдання методом навчання з підкріпленням, який дозволяє знаходити близькі до оптимальних алгоритми керування в результаті взаємодії системи керування з об'єктом керування, використовуючи сигнал підкріплення, що характеризує якість керуючих впливів.

Як сигнал підкріплення використаний відомий квадратичний критерій, що дозволяє врахувати як вимоги до точності, так і до витрат на керування. Пошук керуючих впливів на базі навчання з підкріпленням виконаний з використанням алгоритму ітерацій закону керування. Такий алгоритм реалізований на базі архітектури «виконавець»–«критик». Розглянуто різні варіанти представлення виконавця для реалізації закону керування і критика для отримання значень функції вартості з використанням нейромережевих апроксиматорів. Показано, що точність апроксимації оптимального керування залежить від ряду особливостей, а саме від вдалої структури апроксиматорів, вибору методу поновлення параметрів нейронних мереж, а також параметрів алгоритму навчання.

Досліджений підхід дозволяє вирішувати розглянутий клас задач керування с використанням контролерів з різною структурою, при цьому є можливість уточнення алгоритмів керування в процесі функціонування космічного апарату.

Ключові слова: *керування відносним рухом, космічний апарат з неповним складом виконавчих органів, навчання з підкріпленням, ітерації закону керування, виконавець–критик.*

The aim of the article is to approximate optimal relative control of an underactuated spacecraft using reinforcement learning and to study the influence of various factors on the quality of such a solution.

In the course of this study, methods of theoretical mechanics, control theory, stability theory, machine learning, and computer modeling were used.

The problem of in-plane spacecraft relative control using only control actions applied tangentially to the orbit is considered. This approach makes it possible to reduce the propellant consumption of reactive actuators and to simplify the architecture of the control system. However, in some cases, methods of the classical control theory do not allow one to obtain acceptable results. In this regard, the possibility of solving this problem by reinforcement learning methods has been investigated, which allows designers to find control algorithms close to optimal ones as a result of interactions of the control system with the plant using a reinforcement signal characterizing the quality of control actions.

The well-known quadratic criterion is used as a reinforcement signal, which makes it possible to take into account both the accuracy requirements and the control costs. A search for control actions based on reinforcement learning is made using the policy iteration algorithm. This algorithm is implemented using the actor–critic architecture. Various representations of the actor for control law implementation and the critic for obtaining value function estimates using neural network approximators are considered. It is shown that the optimal control approximation accuracy depends on a number of features, namely, an appropriate structure of the approximators, the neural network parameter updating method, and the learning algorithm parameters.

The investigated approach makes it possible to solve the considered class of control problems for controllers of different structures. Moreover, the approach allows the control system to refine its control algorithms during the spacecraft operation.

Keywords: relative control, underactuated spacecraft, reinforcement learning, policy iteration, actor–critic.

Introduction. Spacecraft (SC) relative motion control is an important task for many space missions. For example, such control is necessary to perform rendezvous and docking of satellites for the delivering astronauts and cargo to a space station [1], maneuvering around a SC for servicing [2], formation of a constellation of satellites of a given configuration [3], contactless removal of space debris [4].

© S. Khoroshylov, M. Redka, 2020

Various aspects of SC dynamics and relative control in circular orbits are considered in Refs. [5, 6]. The peculiarities of solving such a problem in elliptic orbits are presented in articles [7, 8]. Papers [9, 10] investigate the possibility of controlling the in-plane relative motion of a SC using only control actions applied tangentially to the orbit. This approach makes it possible to reduce the propellant consumption of the thrusters and to simplify the architecture of the control system (CS). However, as shown in Ref. [11], for the case of elliptical orbits, such a configuration of the actuators leads to a time-periodic control error when control actions are generated by a linear controller. Reference [12] shows that this error can be reduced by using a time-periodic reference signal generated in a special way. However, the question of optimality of the proposed solution remained open due to the complexity of the analytical solution of the considered problem.

Currently, much attention has been paid in publications to reinforcement learning (RL) methods [13–16], which allow finding control algorithms close to optimal as a result of interaction of the CS with the plant using a reinforcement signal characterizing performance of the control actions. An overview of various RL methods is presented in article [17]. Despite the fact that this approach can be used to solve control problems for arbitrary dynamical objects using controllers with different structures, in practice it is not always possible to obtain an acceptable solution due to the need to choose the structure of the controller and value function approximators and hyperparameters that determine the learning process [18]. In this regard, it is of interest to study the possibility of using the RL methods to find optimal relative control laws for an underactuated SC.

The aim of the article is to approximate optimal relative control of an underactuated SC using RL and to study the influence of various factors on the quality of such a solution.

Equations of motion. The motion of a chief SC (CSC) relative to a deputy SC (DSC) in the orbital plane is considered. It is assumed that only the CSC performs control actions in order to provide the required parameters of the relative motion.

A local-vertical/local-horizontal frame $Oxyz$ (LVLH) is used to determine the position of the CSC with respect to the DSC. The frame origin is at the center of mass of the CSC. The x -axis points along the position vector of the CSC, with respect to the Earth. The z -axis is taken along the direction normal to the plane defined by the orbital position and velocity vectors, and pointing towards the positive values of the orbital angular momentum. The y -axis forms a right-handed coordinate system.

The in-plane relative dynamics for the CSC-DSC formation can be described using the following system of linearized equations [19]:

$$\begin{aligned} \ddot{x} - \omega^2 x - 2\omega\dot{y} - \dot{\omega} y - 2l\dot{x} &= \frac{f_x^d}{m^d} - \frac{f_x^c}{m^c}, \\ \ddot{y} - \omega^2 y + 2\omega\dot{x} + \dot{\omega} x + l\dot{y} &= \frac{f_y^d}{m^d} - \frac{f_y^c}{m^c}, \end{aligned} \tag{1}$$

where x , y are the coordinates of the position vector that represents the position of the center of mass of the DSC with respect to the origin of the LVLH; m^C and m^d are the masses of the CSC and DSC, respectively; f_x^c and f_y^c are the forces

applied to the CSC in the x and y directions, respectively; f_x^d and f_y^d are the forces applied to the SDO in the x and y directions, respectively.

The parameters of Eqs. (1) are calculated as follows:

$$\omega = \sqrt{\frac{Gm}{p^3}}(1+\varepsilon\cos\upsilon),\ p = a\left(1-\varepsilon^2\right),\ \dot{\omega} = -2\varepsilon\sqrt{\frac{Gm}{p^3}}\sin\upsilon(1+\varepsilon\cos\upsilon)\omega,$$

$$I = \frac{Gm}{R^3},\ R = \frac{a\left(1-\varepsilon^2\right)}{1+\varepsilon\cos\upsilon},$$

where Gm is the Earth gravitation constant; υ is the true anomaly; ε is the eccentricity of the orbit; a is the semi-major axis.

The CSC is constantly oriented so that the control force is applied only in y - direction of the LHLV.

Neglecting the effect of external disturbances, the model (1) can be given for the state vector $X(t)=[x, y, \dot{x}, \dot{y}]^T$ and control $u(t)$ using a state-space representation as

$$\dot{X} = AX + Bu\ , \tag{2}$$

where

$$A = \begin{bmatrix} 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \\ \omega^2+2I & \dot{\omega} & 0 & 2\omega \\ -\dot{\omega} & \omega^2-I & -2\omega & 0 \end{bmatrix},\ B = \begin{bmatrix} 0 \\ 0 \\ 0 \\ -1/m^s \end{bmatrix}.$$

A modern SC controller is implemented in a discrete computer system. Therefore, the following discrete form of the model (2) is used:

$$X_{k+1} = A_k X_k + B_k u_k\ , \tag{3}$$

where $A_k = (I + At_s)$; $B_k = Bt_s$; t_s is the sample time; k is the sample number.

We also assume that full state vector is measurable and these measurements are not corrupted by noise.

Discrete-time linear-quadratic regulator. The discrete-time linear-quadratic regulator (DLQR) problem [20] is a widely used methodology to design controllers. The goal of the DLQR design is to find a static gain *K* for the full-state feedback law that minimizes the quadratic cost function:

$$J = \min \sum_{k=0}^{\infty}\left(Q^T X_k Q + R^T u_k R\right), \tag{4}$$

where Q, R are the weight matrices which penalize the system states X_k and the control input u_k, respectively.

Impressive robust stability properties of the DLQR allow designers to use it for systems whose real parameters differ significantly from the nominal ones.

A DLQR implements the full-state feedback law for the CSC in-plane relative control in the following form:

$$u_k = K\left(X^r - X_k\right),$$

where X^r is the reference input vector, which determines the required relative position between the CSC and DSC.

The optimal feedback gain matrix is given by

$$K = \left(R + B^T P B\right)^{-1} B^T P A \ ,$$

where P is the unique positive semi-definite solution of the discrete-time Riccati equation

$$P = Q + A^T P A - A^T P B \left(R + B^T P B\right)^{-1} B^T P A \ .$$

Reinforcement learning. To solve control tasks by RL, it is assumed that the CS learns by analyzing the results of its actions. These results are evaluated using a simple scalar reinforcement signal received from the plant with which the CS interacts. The reinforcement signal, which can be interpreted as a cost, allows an intelligent control system to change its control algorithms in order to achieve a long-term goal.

The general RL algorithm shown in Fig. 1, includes the following steps [21]:

1) at the time moment t_k the plant is in the state X_k;

2) in this state, the CS chooses one of the possible control actions U_k;

3) the CS performs this action, which leads to the transition of the plant to a new state X_{k+1} and receiving the reinforcement C_k;

4) $t \leftarrow t_{k+1}$;

5) go to step 2 or completion if the new state is final.

Let χ is the set of states, and A is the set of control actions. Reinforcement C_k is a consequence of the action U_k chosen in the state. The reinforcement signal is a function that depends on a vector defined in the space $\chi \times A$.

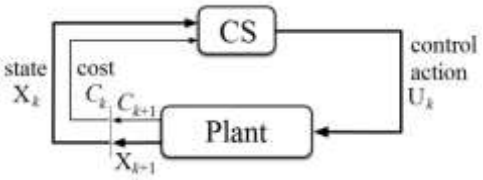


Fig. 1 – RL setup

Total cost can be given as

$$G_k = C_k + \gamma C_{k+1} + \gamma^2 C_{k+2} + \ldots = \sum_{i=0}^{\infty} \gamma^i C_{k+i} \ , \ 0 \le \gamma \le 1 .$$

The discount factor γ determines the degree of significance of future costs for choosing control actions.

One of the key notion of the RL is the value function. Suppose that in each state X_k, the CS generate a control action in accordance with a certain policy π:

$$U_k = \pi(X_k),$$

then the value function determines the total cost that can be paid by going from the initial state X_k and forming control actions in accordance with the policy π. This function can be represented as follows:

$$V^{\pi}(X_k)=\sum_{i=0}^{\infty}\gamma^k C_{k+i}(X_{k+i},U_{k+i})=C_k(X_k,U_k)+\gamma V^{\pi}(X_{k+1}).$$

Actor - critic architecture. There are various algorithms for finding the optimal control using the RL. In this paper, we use the policy iteration algorithm [21] to learn a control law, which has better convergence compared to other algorithms, but less efficient in terms of sample efficiency (amount of data needed for training). Considering that in this study the model of the plant is used for training but not its real transitions, this factor is not so significant.

The essence of this algorithm is to alternately clarify the value function and improve the control law. The algorithm includes the following steps:

1. An initial policy π is selected.

2. The value function V^{π} is estimated for this policy.

3. A certain number of iterations are performed to improve the policy by minimizing the following objective function:

$$\pi(X)\leftarrow \arg\min_U\left[C(X_k,U_k)+\gamma V^{\pi}(X_k)\right].$$

4. Steps 2 and 3 are repeated until the optimal policy π^* and the corresponding to it value function V^{π^*} are obtained.

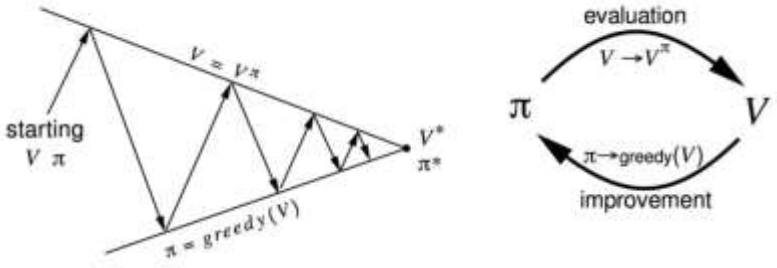


Fig. 2 – Policy iteration algorithm

This algorithm can be implemented using critic and actor modules. In this case, the critic forms the estimates of the value function, and the actor generates control actions.

The critic and actor are implemented in the form of feedforward neural networks (NN), which approximate the cost function and the control law, respectively as follows:

$$V_{\eta}^{\pi}(X),\ \pi_{\theta}(X),$$

where η, θ are the parameter vectors the critic and actor, respectively.

The critic is trained using the method of temporal differences (TD), based on minimizing the TD error, which is calculated as follows [21]:

$$\delta_k=C_k+\gamma V^{\pi}(X_{k+1})-V^{\pi}(X_k).$$

Using this error the loss function of the critic can be represented as:

$$V_\eta^\pi(\mathrm{X}) \leftarrow \arg\min_{V_\eta^\pi}\left[C_k + \gamma V^\pi(\mathrm{X}_{k+1}) - V^\pi(\mathrm{X}_k)\right].$$

The loss function of the actor uses estimates of the critic and is built as follows:

$$\mathrm{U} = \pi_\theta(\mathrm{X}) \leftarrow \arg\min_{\pi_\theta}\left[C(\mathrm{X}_k, \pi_\theta) + \gamma V_\eta^\pi(\mathrm{X}_{k+1})\right].$$

Problem statement and system data. Let us consider the problem of controlling the motion of the CSC relative to the DSC for the following initial data: altitude of the orbit is 640 km; orbital eccentricity is zero; mass of the CSC is 500 kg; mass of the DSC is 1575 kg; maximum control thrust is 0.3 N; the sampling period of the CS is 30 s**.**

Weights of the criteria (4) represented by the following matrices:

$$Q = \mathrm{diag}(0.1,\ 0.1,\ 0,\ 0),\ R = 0.8\,.$$

For this data, the gain matrix of the DLQR has the following values:

$$K = [k_1, k_2, k_3, k_4] = [-0.4214, 0.1926, -257.2522, -15.1899].$$

The gain values for different components of the state vector differ significantly. This can lead to difficulties in training using RL. To eliminate this drawback, the state vector and control are normalized as follows:

$$\overline{\mathrm{X}} = [x/x_n, y/y_n, \dot{x}/\dot{x}_n, \dot{y}/\dot{y}_n]^T,\ \bar{u} = u/u_n,$$

where x_n, y_n, $\dot{x}_n$, $\dot{y}_n$ u_n are the corresponding normalizing values.

For the normalized state vector, the dynamical model takes the following form:

$$\dot{\overline{\mathrm{X}}} = \overline{A}\,\overline{\mathrm{X}} + \overline{B}\bar{u},$$

where $\overline{A} = N^{-1}AN$, $\bar{u} = u_n N^{-1}u$, $N = diag\,(x_n, y_n, \dot{x}_n, \dot{y}_n)$.

The corresponding normalized discrete system is given as:

$$\overline{\mathrm{X}}_{k+1} = \overline{A}_k\,\overline{\mathrm{X}}_k + \overline{B}_k\bar{u}_k, \tag{5}$$

where $\overline{A}_k = (I + \overline{A}t_s)$; $\overline{B}_k = \overline{B}t_s$.

For the normalized system (5), the criterion (4) is written as follows:

$$\bar{J} = \min\sum_{k=0}^{\infty}\left(\overline{Q}^T\,\overline{\mathrm{X}}_k\,\overline{Q} + \overline{R}^T\bar{u}_k\,\overline{R}\right),$$

where $\overline{Q} = N^{-1}QN$, $\overline{R} = u_n^2 R$.

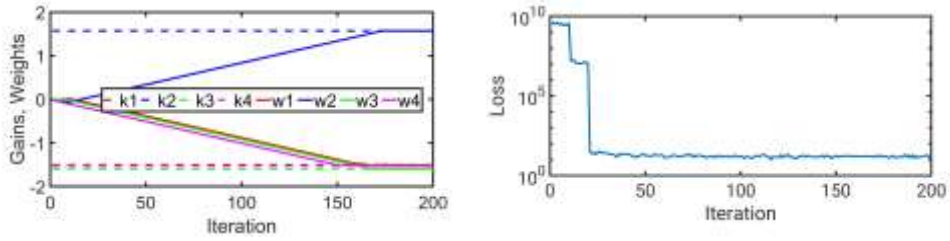
For the following normalizing values $x_n = 1.05$ m, $y_n = 2.5$ m, $\dot{x}_n = 18\cdot 10^{-4}$ m/s, $\dot{y}_n = 2.4\cdot 10^{-2}$ m/s, $u_n = 0.3$ H, the gain matrix of the normalized DLQR has the following values $\overline{K} = [-1.5026, 1.5730, -1.5897, 1.5188]$.

Further, considering this result as a baseline controller, we investigate the possibility of obtaining the same result but using the RL.

Numerical experiments.

RL1 case. In this case, a NN approximator of the actor is used. The NN includes an input feature layer, the dimension of which is equal to the dimension of the state vector. One fully connected layer and an output layer with a dimension equals to the dimension of the control vector. The estimates of the value function are calculated directly using the model without using an approximator. Stochastic gradient descent (SGD) [22] with a mini-batch size of 64 and a learning rate of 0.01 was used to update the weights of the NN actor. The weights of the actor had been fixed every 10 iterations when the value function was calculated.

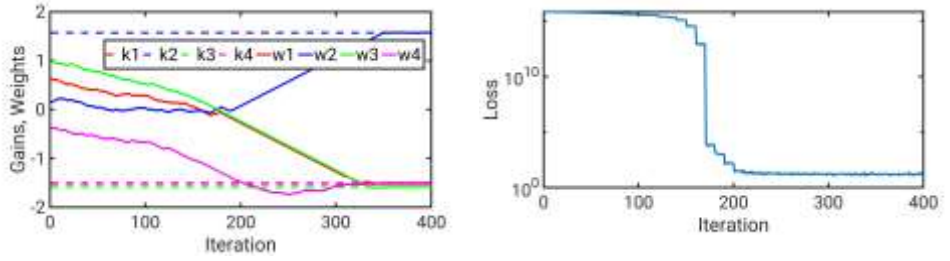
Figure 3 shows the learning process for such a case when the weights of the actor are initialized by normally distributed random numbers with zero mean and a standard deviation of 0.01. As can be seen from this figure, the values of the weights of the actor converge to the corresponding gains of the normalized DLQR (plotted by dashed lines). As a consequence, the total cost for this case is equal to the cost for the DLQR case (see Table 1).



a) b)

Fig. 3 – Learning process for the RL 1 case when the weights of the actor is initialized by small random numbers (a – weights of the actor, b – loss function)

The process of training the actor using Xavier initialization [22] is shown in Fig. 4. As can be seen in this case, the values of the weights also converge to their optimal values, but the learning process requires about twice as many iterations as in the case with small number initialization.



a) b)

Fig.4 – Learning process for the RL 1 case when Xavier initialization is used (a – weights of the actor, b – loss function)

RL 2 case. In this case, the tanh activation function is used at the output of the NN actor to limit the magnitude of the control actions. As can be seen in Fig. 5, in this case, the values of the NN weights slightly exceed the values of the DLQR gains, but the total cost for this case practically does not differ from the optimal one.

The next experiments were carried out using the NN critic to approximate the value function. The NN of the critic includes a feature input layer, a quadratic layer, a fully connected layer, and an output layer with a dimension of 1. In this case, the optimal values of the weights of the critic are not known. Therefore, to obtain the baseline values of the weights, the critic was trained using supervised learning. To implement this, the target values of the value function were calculated using the model of the plant.

As can be seen in Fig. 6, the structure of the NN allows the cost function to be approximated pretty accurate.

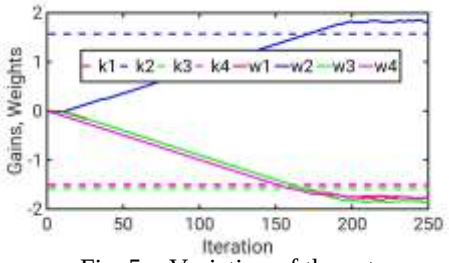


Fig. 5 – Variation of the actor weights for the RL 2 case

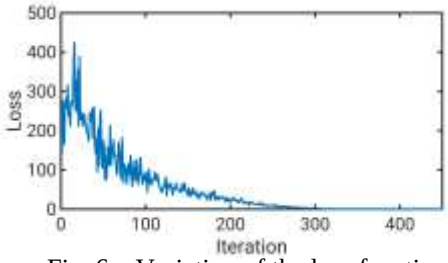
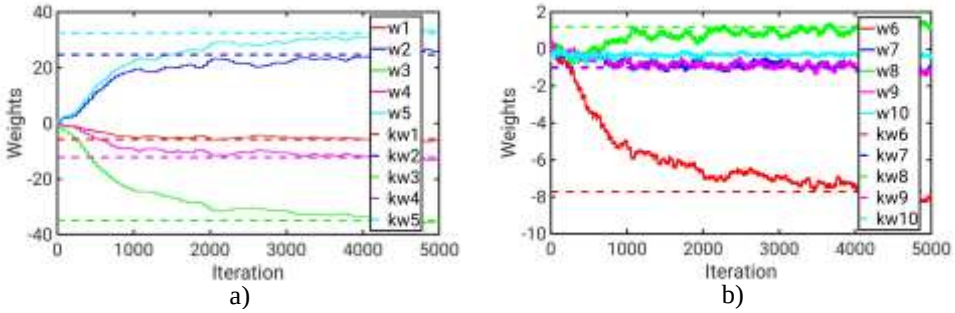


Fig. 6 – Variation of the loss function during supervised learning

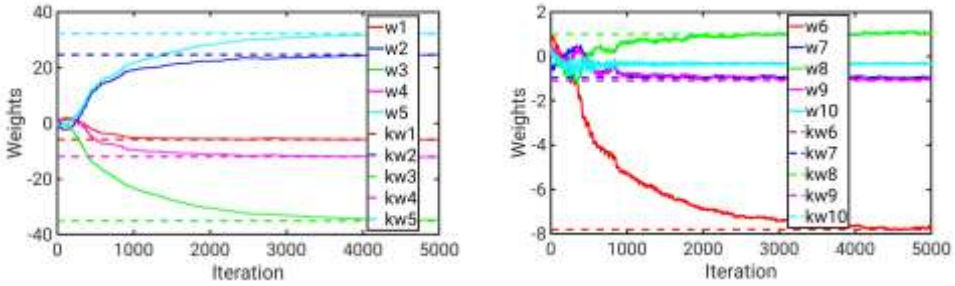
Figure 7 shows the process of updating the weights of the critic using the TD method. To update the weights, the SGD was used with a learning rate of 0.075 and the number of iterations for updating the target was equal to 50. As can be seen from this figure, in this case there is a noticeable deviation of the weights of the critic from the baseline values (dashed lines).



a) b)

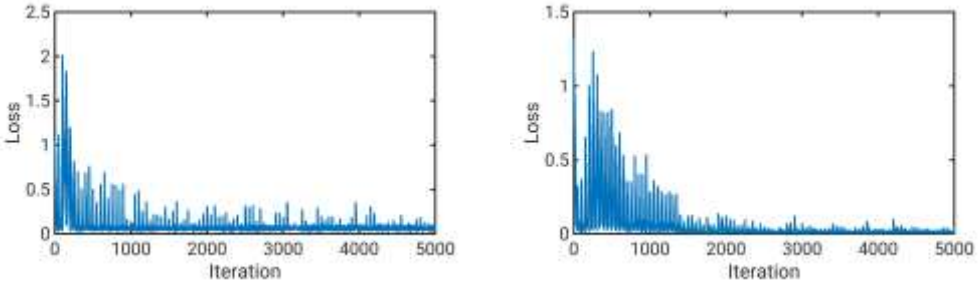
Fig. 7 – Critic weight updates using SGD (a – w1-w5, b – w6-w10)

The ADAM optimizer with the parameters 'Gradient Decay Factor' = 0.9 and 'Squared Gradient Decay Factor' = 0.999 significantly increases the accuracy of the value function approximation (see Fig. 8, 9) in comparison with SGD.



a) b)

Fig. 8 – Critic weight updates using ADAM (a – w1-w5, b – w6-w10)



a) b)

Fig. 9 – Loss function variation (a – SGD, b – ADAM)

RL 3 case. In this case, the NNs are used both for the critic and for the actor. To update the weights, the SGD was applied with a mini-batch size of 64 and a learning rate of 0.01 and 0.075 for the actor and critic, respectively. The weights were repeatedly updated in the loop using 50 and 10 iterations for the critic and actor, respectively. As can be seen in Figs. 10–12, the approximation error of the value function leads to a noticeable variation in the weights of the actor. Nevertheless, such learning errors of the actor do not lead to significant deviations of the control actions and the trajectory of motion from the optimal one (see Figs. 13–14).

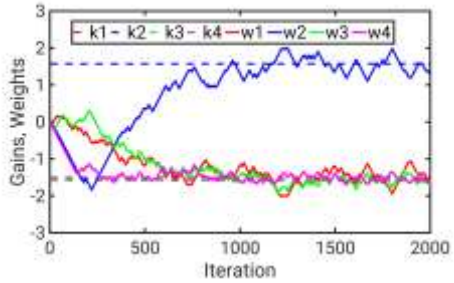


Fig. 10 – Variation of the actor weights for the RL 3 case

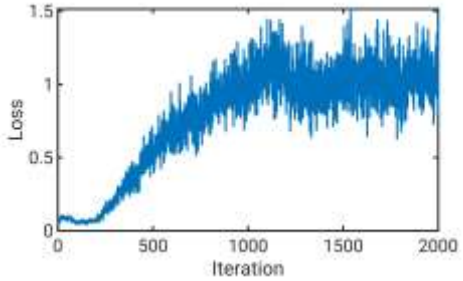
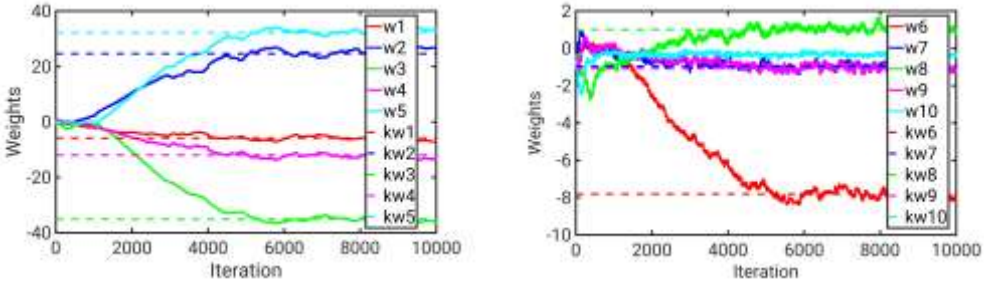


Fig. 11 – Variation of the actor loss function for the RL 3 case



a) b)

Fig. 12 – Variation of the critic weights for the RL 3 case (a – w1-w5, b – w6-w10)

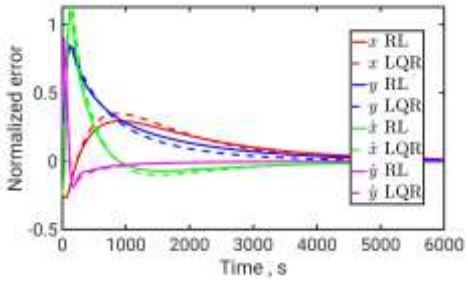


Fig. 13 – Variation of the normalized errors of the SC relative motion for the RL 3 case

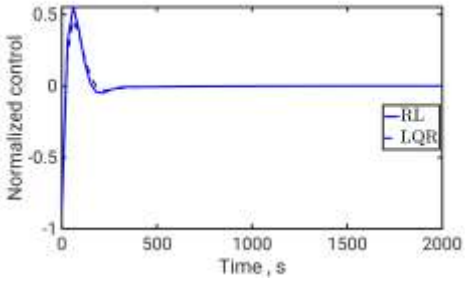


Fig. 14 – Variation of the normalized control actions for the RL 3 case

RL 4 case.

In this case the ADAM optimizer was used instead of the SGD to update the weights of the critic. This makes it possible to approximate the value function more accurately (Fig. 17) in comparison with the previous case (see Fig. 14). This allowed us to obtain results that are closer to optimal ones (Figs. 15, 16).

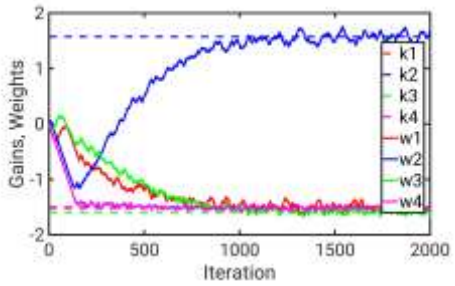


Fig. 15 – Variation of the actor weights for the RL 4

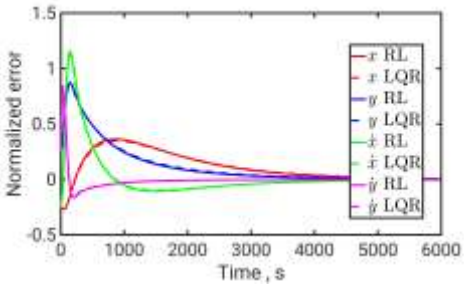
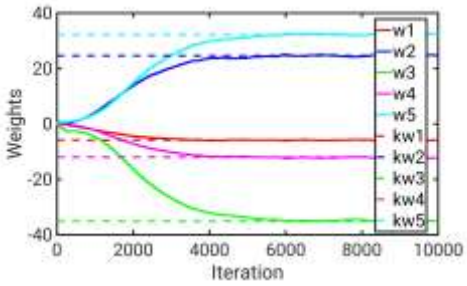
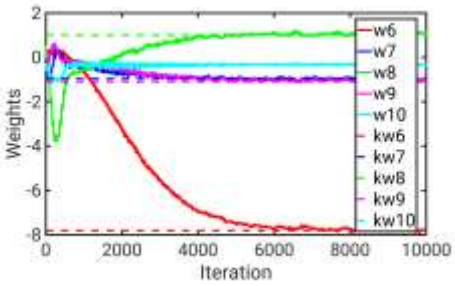


Fig. 16 – Variation of the normalized errors of the SC relative motion for the RL 4



a)



b)

Fig. 17 – Variation of the critic weights for the RL 4 case (a – w1-w5, b – w6-w10)

RL 5 case.

Figures 18–20 show the results when the ADAM method is used to train both the actor and the critic. However, as can be seen from these figures and Table 1, this approach does not provide better results than in the RL 4 case.

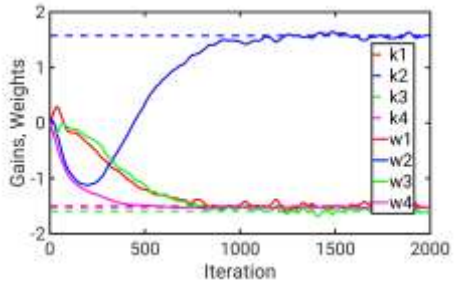


Fig. 18 – Variation of the actor weights for the RL 5 case

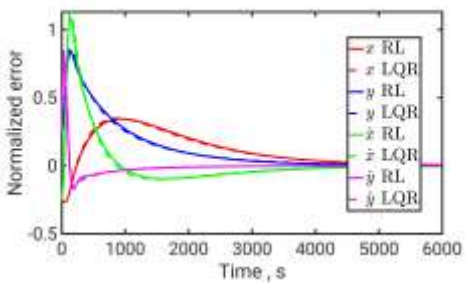
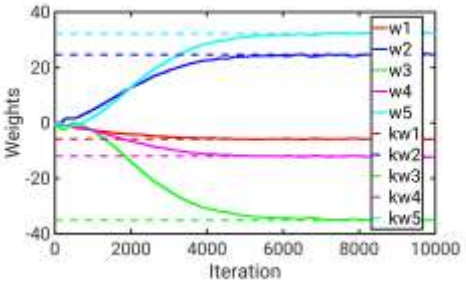
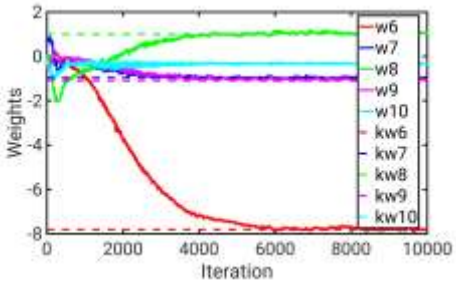


Fig. 19 – Variation of the normalized errors of the SC relative motion for the RL 5



a)



b)

Fig. 20 – Variation of the critic weights for the RL 5 case (a – w1-w5, b – w6-w10)

Table 1 shows the total costs obtained for the considered cases when the CSC moves from the initial state $\overline{X} = [-0.25, 0.25, -0.25, 0.25]^T$ during 200 sample periods.

Table 1. Total costs.

Case	DLQR	RL 1	RL 2	RL 3	RL 4	RL 5
Total cost	7.1908	7.1908	7.1955	7.3040	7.1970	7.1994

Conclusion. The article demonstrates the possibility of accurate approximation of the optimal control of the relative motion of an underactuated spacecraft in circular orbits using reinforcement learning. It is shown that in this case control performance depends on a number of features, namely, the correct structure of the approximators, type of optimizers, and hyperparameters of the learning algorithms.

The investigated algorithm can be used to find aproximate optimal control of the relative motion of the underactuated spacecraft in elliptical orbits, which may be a subject of the future studies.

1. *MacIsaac D.* Docking at the International Space Station. Phys. Teach. 2014. Vol. 52. No126. https://doi.org/10.1119/1.4862134
2. *Campbell M., Fullmer R.R., Hall C.D.* The ION-F formation flying experiments. Advances in the Astronautical Sciences. 2000. Vol. 105. P. 135–149.
3. *Smith G.W., DeRocher W.L. Jr.* Orbital servicing and remotely manned systems. Mechanism and Machine Theory. 1977. Vol. 12. P. 65–76. https://doi.org/10.1016/0094-114X(77)90058-1
4. *Alpatov A.P., Khoroshylov S.V., Maslova A.I.* Contactless de-orbiting of space debris by the ion beam. Dynamics and control. Kyiv: Akademperiodyka, 2019. 170 p. https://doi.org/10.15407/akademperiodyka.383.170
5. *Vassar R.H., Sherwood R.B.* Formationkeeping for a pair of satellites in a circular orbit. Journal of Guidance, Control, and Dynamics. 1985. Vol. 8(2). P. 235–242. https://doi.org/10.2514/3.19965
6. *Redding D.C., Adams N.J., Kubiak E.T.* Linear quadratic stationkeeping for the STS orbiter. Charles Stark Draper Laboratory, Cambridge, MA, Kept. CSDL-R-1879, June 1986. https://doi.org/10.2514/6.1986-2222

7. *Dwidar H.R., Owis A.H.* Relative Motion of Formation Flying with Elliptical Reference Orbit. International Journal of Advanced Research in Artificial Intelligence. 2013. Vol. 2(6). P. 79–86. https://doi.org/10.14569/IJARAI.2013.020613
8. *Peng H., Zhao J., Wu Z., Zhong W.* Optimal periodic controller for formation flying on libration point orbits. Acta Astronaut. 2011. Vol. 69. P. 537–550. https://doi.org/10.1016/j.actaastro.2011.04.020
9. *Starin R.S., Yedavalli R.K., Sparks A.G.* Spacecraft formation flying maneuvers using linear-quadratic regulation with no radial axis inputs. AIAA Paper. August 2001. P. 2001–4029. https://doi.org/10.2514/6.2001-4029
10. *Kumara K.D., Bang H.C., Tahk M.J.* Satellite formation flying using along-track thrust. Acta Astronautica. 2007. Vol. 61 (7-8). P. 553–564. https://doi.org/10.1016/j.actaastro.2007.01.069
11. *Alpatov A., Khoroshylov S., Bombardelli C.*, Relative Control of an Ion Beam Shepherd Satellite Using the Impulse Compensation Thruster. Acta Astronautica. 2018. Vol. 151. P. 543–554. https://doi.org/10.1016/j.actaastro.2018.06.056
12. *Khoroshylov S.* Relative control of an ion beam shepherd satellite in eccentric orbits. Acta Astronautica. 2020. Vol. 176. P. 89–98. https://doi.org/10.1016/j.actaastro.2020.06.027
13. *Haarnoja T., Zhou A., Abbeel P., Levine S.* Soft actor-critic: Off-policy maximum entropy deep reinforcement learning with a stochastic actor. 2018. arXiv preprint arXiv:1801.01290.
14. *Jaderberg M., Mnih V., Czarnecki W.M., Schaul T., Leibo J.Z., Silver D., Kavukcuoglu K.* Reinforcement learning with unsupervised auxiliary tasks. 2016 arXiv preprint arXiv:1611.05397.
15. *Khadka S., Tumer K.* Evolution-guided policy gradient in reinforcement learning. Advances in Neural Information Processing Systems. 2018. P. 1196–1208.
16. *Nair A., McGrew B.* Andrychowicz M., Zaremba W., Abbeel P. Overcoming exploration in reinforcement learning with demonstrations. 2018 IEEE International Conference on Robotics and Automation (ICRA) .2018. P. 6292–6299. https://doi.org/10.1109/ICRA.2018.8463162
17. *Kober J., Bagnell J. A., Peters J.* Reinforcement learning in robotics: A survey. International Journal of Robotic Research. 2013. Vol. 32(11). P. 1238–1274. https://doi.org/10.1177/0278364913495721
18. *Khoroshylov S. V., Redka M. O.* Intelligent control of spacecraft attitude using reinforcement leaning. Technical Mechanics. 2019. No 4. P. 29–43. (in Ukrainian). https://doi.org/10.15407/itm2019.04.029
19. *Yamanaka K., Ankersen F.* New State Transition Matrix for Relative Motion on an Arbitrary Elliptical Orbit. Journal of Guidance, Control, and Dynamics. 2002. Vol. 25 (1). P. 60–66. https://doi.org/10.2514/2.4875
20. *Lewis F.L., Vrabie D., Syrmos V.L.* Optimal Control, 3rd Edition. NY: John Wiley & Sons, Inc., 2012. https://doi.org/10.1002/9781118122631
21. *Sutton R.S., Barto A.G.* Reinforcement learning: an introduction. MIT press, 1998. 338 p.
22. *Glorot X., Bengio Y.* Understanding the difficulty of training deep feedforward neural networks. Proceedings of the thirteenth international conference on artificial intelligence and statistics. 2010. P. 249–256.

Received on 09.11.2020,
in final form on 23.11.2020