UDK 629.5 https://doi.org/10.15407/itm2024.04.063

S. V. KHOROSHYLOV, V. K. SHAMAKHANOV

DEPLOYMENT CONTROL OF TRANSFORMABLE ROD STRUCTURES USING REINFORCEMENT LEARNING

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

Розглянуто завдання керування розгортанням стрижневих конструкцій космічного застосування, що трансформуються. Прикладом таких конструкцій є ферма сітчатої антени, яка розгортається за допомого системи з тросів та шківів.

Метою дослідження є розробка на основі методології навчання з підкріпленням (НЗП) інтелектуального агенту (ІА), який забезпечує розгортання та утримання в розгорнутому положенні розглянутої конструкції з урахуванням заданих вимог. Основними вимогами є час розгортання та мінімальні кутові швидкості стрижнів V-подібного складання на кінцевому етапі розгортання конструкції.

Під час проведення досліджень використано методи моделювання динаміки систем зв'язаних тіл, теорії керування, навчання з підкріпленням та комп'ютерного моделювання.

Продемонстровано можливість використання методології НЗП для подолання низки складнощів, притаманних традиційним підходам при керуванні розгортанням стрижневих конструкцій, що трансформуються. Зокрема, НЗП дає змогу оптимізувати систему розгортання з використанням моделей, отриманих за допомогою спеціалізованого програмного забезпечення для моделювання динаміки систем зв'язаних тіл, враховуючи необхідні критерії та обмеження.

Особливості використання такого підходу для керування розгортанням стрижневих конструкцій досліджено з використанням спрощеної моделі однієї секції сітчатої антени, що трансформується. ІА побудовано на базі архітектури виконавець–критик. Запропонована структура нейронних мереж ІА, що забезпечують реалізацію обмежень на керуючі впливи та стійкість процесу навчання. При навчанні ІА застосовано алгоритм оптимізації найближчих політик. Розглянуто різні випадки, що відрізняються функціями вартості, функціями активації виконавця, параметрами тертя в шарнірах.

У тих випадках, коли динамічні властивості моделі та реальної структури суттєво відрізняються, ІА можливо довчити. Ця операція може бути реалізована шляхом розгортання реальної структури, оскільки ІА вимагає значно менше спроб для остаточного точного налаштування, ніж для попереднього навчання.

Практична цінність отриманих результатів полягає в тому, що вони дозволяють пришвидшити розробку систем керування розгортанням космічних конструкцій та підвищити якість цих процесів з урахуванням необхідних критеріїв.

***Ключові слова**: конструкція, що трансформується; навчання з підкріпленням; нейронна мережа; керування розгортанням.*

The task of controlling the deployment of transformable rod structures for space applications is studied. An example of such structures is a mesh antenna truss, which is deployed using a cable-pulley system.

The aim of the study is to develop an intelligent agent (IA) based on the reinforcement learning (RL) methodology, which ensures the deployment and maintenance of the structure under consideration in the deployed position, taking into account the specified requirements. The main requirements are the deployment time and the minimum angular velocities of the V-folding rods at the final stage of the structure deployment.

During the research, methods of dynamic modeling of multibody systems, control theory, reinforcement learning, and computer simulation were used.

The possibility of using the RL methodology to overcome a number of difficulties inherent in traditional approaches to controlling the deployment of transformable rod structures is demonstrated. In particular, the RL allows optimizing the deployment system using models obtained using specialized software for modeling of the multibody dynamics, taking into account the necessary criteria and constraints.

The features of this approach to controling the deployment of rod structures were investigated using a simplified model of one section of a transformable mesh antenna. The AI was designed on the basis of the actor-critic architecture. The structure of AI neural networks was proposed, which ensure the implementation of

constraints on control actions and the stability of the learning process. Proximal policy optimization algorithm is used for training the IA. Various cases are investigated, which differ in cost functions, actor activation functions, and friction parameters of the joints.

In cases where the dynamic properties of the model and the real structure differ significantly, the AI can be fine-tuned. This operation can be implemented by deploying the real structure, since the AI requires significantly fewer attempts for final fine-tuning than for preliminary training.

The practical value of the obtained results is that they allow facilitating the development of space structure deployment control systems and improve their performance according to different specified criteria.

Keywords: *transformable structure; reinforcement learning; neural network; deployment control.*

Introduction. Investigation of autonomous deployable lightweight structures has emerged as a leading focus in aerospace engineering in recent years. A deployable lightweight structural system fundamentally consists of an integrated structure and mechanisms. It can be easily transported and stored in a compact, stowed state, while allowing for a considerably larger operational configuration once deployed. These features have garnered significant attention from numerous researchers in the field of deployable lightweight structures.

Mesh reflector antennas have been extensively utilized in space applications due to their different advantages such as a large aperture size, minimal total mass, compact stowed volume, and reduced surface distortion [1–4]. The antenna transitions from a stowed state to a fully deployed position, ultimately creating the necessary functional surface. This deployment process significantly influences the performance of antennas in orbit.

Structural deployment is typically executed through various active control mechanisms, utilizing different types of actuators such as active struts and cables, to ensure a rapid and secure deployment process.

The process of the antenna deployment is inherently complex, involving both mechanical and structural considerations, and is susceptible to potential malfunctions [5]. It is essential that the angular speed and acceleration during deployment are controlled to ensure a smooth operation. Additionally, the angular acceleration must remain within specified limits to prevent excessive impact, which could result in vibrations or damage of the antenna [6]. Consequently, it is crucial to develop an effective control algorithm that facilitates precise and smooth deployment.

Efficient and precise deployment in orbit is essential for the proper functioning of antennas. A well-defined deployment strategy is crucial to establish the kinematic behavior of the deployment process and the loading characteristics of the driving force [7]. Additionally, a satellite system's energy needs encompass power for efficient load management, communication, and the maintenance of satellite attitude.

For satellites equipped with large deployment antennas, the energy required for antenna deployment is a crucial factor. Therefore, the antenna deployment should be designed in such a way as to minimize the deployment impact on the structure to limit the peak power required for the deployment mechanism.

In Ref [8], a decoupling control approach is introduced for the precise deployment of space flexible antennas. The rigid and flexible controllers are developed based on the distinct characteristics of the decoupled feedback. The rigid controller guarantees that the antenna follows a predetermined trajectory, while the flexible controller mitigates flexural vibrations.

A force-controlled approach is introduced in Ref [9] and the relationship between the driving force and the deployment motion of the reflector is established. The driving force variations are determined based on the planned deployment motion. The deployment dynamics of the deployable mesh antenna are simulated, and the influences of initial velocity, damping, and gravity on the deployment process are analyzed.

The interdependent relationship between the antenna structure, deployment trajectory, and control system is examined in Ref [10]. A multi-objective function is established to concurrently minimize the antenna's mass, the impacts on the antenna, and the energy dissipation within the control system. The design variables are identified as the cross-sectional areas of the links, Bezier control points, and controller gain parameters.

An optimization approach for the winding strategy of the driving cable is introduced in Ref [11] for an AstroMesh-type antenna. The driving force is derived from principles of energy conservation, considering the influences of the cable nets and friction. An optimization model is developed with the goal to minimize the power required for deployment.

A symplectic instantaneous optimal control approach is proposed in Ref [12] for the deployment of structures utilizing sliding cable actuators. The initial continuous control task is transformed into a sequence of constrained symplectic instantaneous optimal control problems at each time interval, ensuring compliance with the input saturation inequality constraints.

Despite a significant progress in the field of the deployment control of rod structures, the use of the approaches described above causes significant difficulties when applied to complex structures, the model of which is obtained using software packages for modeling the dynamics of multibody systems. In addition, these results do not offer a way to further adjust the deployment algorithms considering the difference in the dynamic properties of the model and the real structure.

Currently, deep learning methods [13] are successfully used for various control tasks in space [14, 15]. For such tasks, both supervised learning [16] and reinforcement learning (RL) [17–19] methods are utilized. The latter group of methods allows obtaining control laws by applying a sequence of control actions to the plant, which can be implemented using either a model or a real structure. Given the potential of deep learning and the noted problems in applying conventional methods, it is of interest to analyze the feasibility of using RL to control the deployment of rod structures for space applications.

Problem statement. A mesh antenna (Fig. 1) from Ref [20] is considered as a transformable structure in this study. The reflective mesh (2) of this antenna is connected to a cable network framework, which is held in tension through a deployable ring truss (1) and tension ties. A cable-pulley system (CPS) is employed to transform the energy generated by electric motors (3) into the driving forces for the truss deployment (Fig. 2).

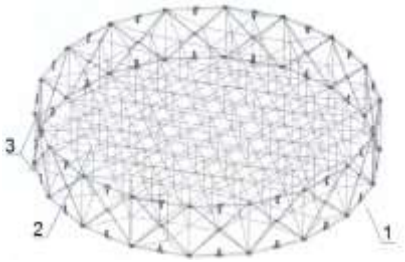


Fig. 1 – Mesh antenna

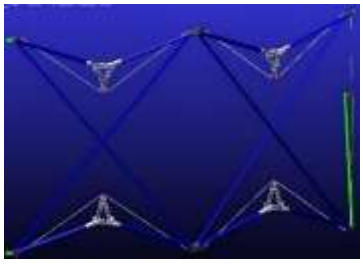


Fig. 2 – Driving cable-pulley system

Using the absolute nodal coordinate formulation (ANCF) the dynamic equations for the whole structure can be expressed as a set of differential-algebraic equations with a constant mass matrix as follows [21]

$$M(X,t)\ddot{X} + D(X,\dot{X},T) + C(X,t) + \hat{O}_{\tilde{O}}\Lambda = V(t), \tag{1}$$

$$\hat{O}(\tilde{O}) = 0, \tag{2}$$

where M is a constant mass matrix of the system; q is the generalized coordinates of the whole multibody system; $Q(q)$ is the elastic force vector of the flexible bodies; $\Phi(q,t)$ is the constraint vector of the system; Φ_q is the derivative matrix of the constraint vector with respect to the generalized coordinates q; Λ is the Lagrange multiplier vector; $F(q,\dot{q})$ is the generalized external force vector; $Q(q)$ is the Jacobian of the elastic force, d is the damping coefficient.

The model (1, 2) is a large dimensional system and its deriviation is a cumbersome task. To facilitate such tasks, specialized software is used for multibody dynamics simulations [22, 23]. Such software has tools that allow finding control actions that ensure the motion of the system along a specified trajectory. Figure 4 shows the variations of the control torques in the hinges of the V-folding rods found using such a tool, which ensure the deployment of the structure with a parabolic variation of the angular velocities (Fig. 3). However, the practical implementation of such control using a CPS is not possible, since it cannot apply torques of different signs.

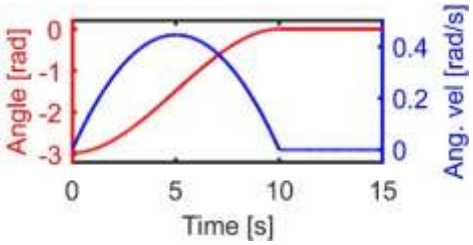


Fig. 3 – State variation

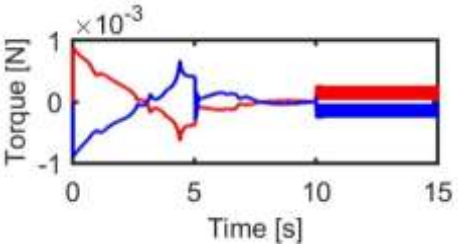


Fig. 4 – Torque variation

When using such a deployment system, it is necessary to solve an optimization problem considering constraints on control actions. Existing algorithms for solving such a problem require a plant model in the following form [24]:

$$q_{k+1} = f(q_k, U_k), \tag{3}$$

where k is the sample number of the discrete system.

However, the software for multibody dynamics simulations provides a model in the following form:

$$q_{k+1} = f\left(q_0, U_0, U_1, U_2, ..., U_k\right). \quad (4)$$

Thus, before finding the optimal control U_k that transfers the system from the state q_k to q_{k+1}, it is necessary to find some sequence of control actions $U_0, U_1, U_2, ..., U_{k-1}$ that ensures the system motion from the initial state q_0 to the required q_k. This feature complicates the application of conventional methodology. This difficulty can be overcome using the RL methodology, since it is based on the analysis of the following Markov decision process

$$q_0, U_0, q_1, C_1; U_1, q_2 C_2; ...; U_{n-1}, q_n C_n.$$

As a result of the RL, it is necessary to find such a sequence of actions U_i that minimize cumulative cost $\sum_{i=1}^{n} C_i$ of completing the task. The cost C_i here is a value of the selected optimality criterion.

To study the possibility of using such an approach to control the deployment of rod structures, we consider a simplified model of one section of a transformable antenna (Fig. 5). All rods of the structure are modeled as rigid bodies. The impacts of the cable deployment system are modeled as identical torques in the hinges of the V-folding rods, and the values of these torque can be only take positive. In addition to the control torques, the torques of viscous friction are applied in the hinges. The model also takes into account constraints on the maximum angle between the V-folding rods.

Fig. 5 – Simplified model.

This model is built using the open source package HotInt [22].

The aim of this study is to develop an RL-based intelligent agent (IA), which ensures the deployment and maintenance in the deployed position of the considered structure taking into account the specified requirements. The main requirements are the deployment time and the minimum angular velocities of V-folding rods at the final stage of deployment.

Table 1. Parameters of the structure.

Parameter	Value	Units
Section height	0.63	m
Length of the diagonal rod	0.4275	m
Outer diameter of the diagonal rod	0.01	m
Inner diameter of the diagonal rod	0.00915	m
Length of the horizontal rod	0.2889	m
Outer diameter of the horizontal rod	0.012	m
Inner diameter of the horizontal rod	0.01115	m
Rod density	1800	kg/m^3

Reinforcement learning based control. The RL control framework operates under the premise that the control system acquires knowledge by examining the outcomes of its actions [25]. These outcomes are assessed through a scalar signal known as reinforcement, which is provided by the plant that the control system engages with. This reinforcement signal serves as a benchmark, enabling the intelligent control system to adjust its control algorithms in light of progress toward achieving long-term objectives.

A general RL algorithm is illustrated in Fig. 2 and consists of the following steps:

1) At time t_k , the system is in state X_k ;

2) In this state, the control system chooses one of the available control actions U_k ;

3) The control system executes this action, resulting in the system transitioning to a new state X_{k+1} , while also receiving a reinforcement signal C_k ;

4) The algorithm then either continues from step 2, incorporating the received reinforcement, or terminates if the new state is designated as final.

We denote χ as a set of states, and A as a set of control actions. Then the reinforcement C_k is a consequence of the action U_k selected in the state X_k . The reinforcement signal is a function that depends on a vector defined in the space $\chi \times A$.

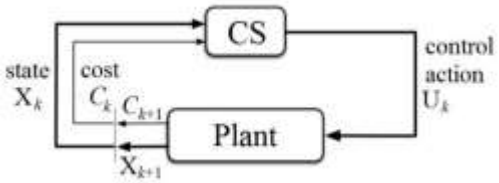


Fig. 6 – Reinforcement learning setup

The control system selects actions aimed at reducing the overall cost, which is calculated in the following manner:

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

The discount factor γ plays a crucial role in assessing the significance of the predicted cost values when choosing control actions. A fundamental component of

RL is a value function. Consider that in each state X_k, the system controller (SC) implements a control action based on a specific algorithm known as a policy π

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

then the value function calculates the overall cost incurred when transitioning from the initial state X_k by choosing control actions in accordance with the policy π. This function can be expressed as:

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

Reinforcement learning can be executed through an actor-critic framework. In this setup, the critic estimates the value function for each state, while the actor translates the state vector into corresponding control actions.

In the framework of deep RL, the actor and critic are represented as feedforward multilayer neural networks, which serve to approximate the control policy and the cost function, respectively:

$$V^{\pi}(X_k, \phi), \pi(X_k, \phi),$$

where θ, ϕ are the vectors of critic and actor parameters, respectively.

This research employs the Proximal Policy Optimization (PPO) algorithm [26]. The implementation of this algorithm is carried out as follows:

1. To determine the total cost of G_t as the sum of the cost for this time step and the discounted future cost [27]

$$G_t = \sum_{k=t}^{ts+m} \left(\gamma^{k-t} C_k\right) + b\gamma^{N-t+1} V(X_{ts+N}, \theta),$$

where b equals 0 when X_{ts+N} represents the final state, and equals 1 in all other cases. In other words, when X_{ts+N} is not the final state, the discounted future value incorporates a function of the discounted state value, which is determined using the critic neural network V.

2. To find the advantage function D_t

$$D_t = G_t - V(X_t, \theta).$$

3. To adjust the critic parameters by minimizing the loss function L_{critic} across all received mini-batch data.

$$L_{critic}(\theta) = \frac{1}{M} \sum_{i=1}^{M} (G_i - V(X_i, \theta))^2.$$

4. To update the actor parameters by minimizing the actor loss function L_{actor} of all received mini-batch data as follows

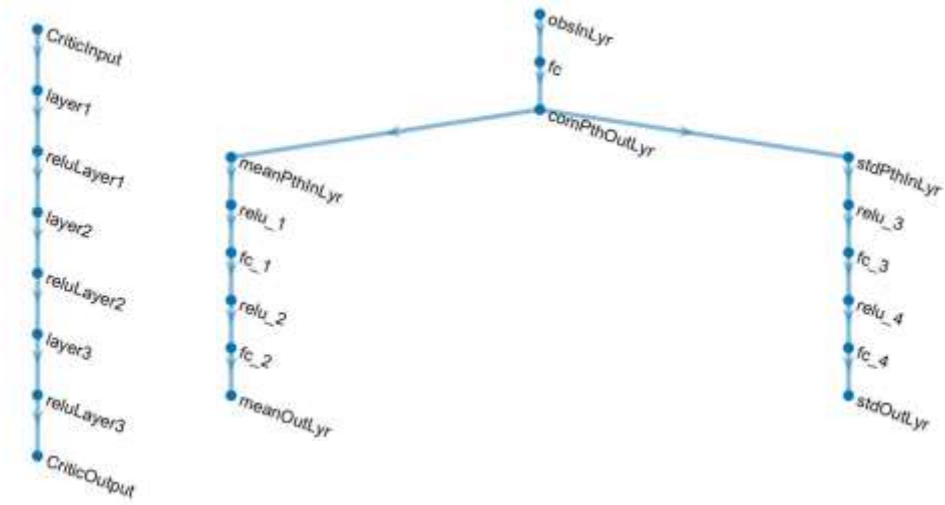
$$L_{actor}(\phi) = \frac{1}{M} \sum_{i=1}^{M} (-\min(r_i(\phi) \cdot D_i, c_i(\phi) \cdot D_i)),$$

$$r_i(\phi) = \frac{\pi(U_i \mid X_i, \phi)}{\pi(U_i \mid X_i, \phi_{old})},$$

$$c_i(\phi) = \max\left(\min\left(r_i(\phi), 1+\varepsilon\right), 1-\varepsilon\right),$$

where D_i and G_i are the advantage and total cost function for the *i*-th element of the mini-batch, respectively; $\pi(U_i \mid X_i, \phi)$ is the probability of executing the action U_i in the state X_i, given the updated policy parameters ϕ; $\pi(U_i \mid X_i, \phi_{old})$ is the probability of action U_i in state X_i, given the previous policy parameters ϕ_{old} prior to the current learning epoch; ε is the clipping parameter.

The actor and critic are implemented in a form of artificial neural networks (NN), the architectures of which presented in Fig. 7. Since the AI agent behave stochastically during training the actor outputs mean value and standard deviation of the control actions. The AI receives the following state vector $X_i = [\varphi, \dot{\varphi}]^T$, where is the angle the V-folding rods.



a b

Fig. 7 – Critic and actor NN architecture

Table 2 and 3 summarize information about neuron numbers used in each layer of the NNs.

Table. 2. Number of neurons in FC layers of the critic

Layer	CriticInput	Layer1	Layer2	Layer3	CriticOutput
Number of neurons	2	20	14	10	1

Table. 3. Number of neurons in FC layers of the actor

Layer	obsInLyr	fc	meanPthInLyr	fc_1	fc_2	stdPthInLyr	fc_3	fc_4
Number of neurons	2	32	32	16	1	32	16	1

The following cost functions are used for training the AI:

Cost 1

if $[((t \le t_d) \text{ and } (\varphi \ge \varphi_d)) \text{ or } ((t > t_d) \text{ and } (\varphi < \varphi_d))] > 0$ then $C_k = 1$ else $C_k = -1$, where t_d is the deployment time; φ_d is the V-folding rod latching angle;

Cost 2

$if\ [((t \le t_d)and\ (\varphi \ge \varphi_d))or\ ((t > t_d)and\ (\varphi < \varphi_d))] > 0$ then $C_k = 1 - U_k^T R U_k$

else $C_k = -1 - U_k^T R U_k$,

where R is the control action weight;

Cost 3

$C_k = 1 - X_k^T R X_k - U_k^T R U_k$ $if\ t \le t_p$ then $Q = Q_1, R = R_1$ else $Q = Q_2, R = R_2$

where t_p is the time, when the weights switch from Q_1, R_1 to Q_2, R_2.

When Cost 1 is used the IA receives +1 reward if the structure deploys during the specified time and -1 penalty on that intervals when time deployment requirements are violated. Cost 2 similar to Cost 1, but also penalizes control actions. Cost 3 is a quadratic criterion with switching weights that penalize state errors and control actions.

Simulation results. To study the feasibility of the RL approach to control the deployment of rod structures, various cases presented in Table 4 are considered. These cases differ in the cost functions, friction coefficients in the hinges of the V-folding rods, and actor activation functions. The nominal deployment time for all cases is 10 s.

Table. 4 – Case description

Case No	Cost	Friction coefficient, [Nms]	Actor output		Q_1	Q_2	R_1	R_2	t_p, c
			Mean	Standard Deviation					
1	1	0	Tanh+ scaling(0.5)	Tanh+ scaling(0.1)	-	-	-	-	-
2	2	0	Tanh+ scaling(0.5)	Tanh+ scaling(0.1)	-	-	1	1	-
3	3	0.001	Tanh+ scaling(0.5)	Tanh+ scaling(0.1)	diag [1,1]	diag [1;8]	1	1	8
4	3	0.0005	Tanh+ scaling(0.5)	Tanh+ scaling(0.1)	diag [1,1]	diag [1;8]	1	1	8
5	3	0.0015	Tanh+ scaling(0.5)	Tanh+ scaling(0.1)	diag [1,1]	diag [1;8]	1	1	8
6	3	0.001	Tanh+ scaling(0.5)	Tanh+ scaling(0.05)	diag [1,1]	diag [1;8]	1	1	8
7	3	0.0015	Tanh+ scaling(0.5)	SoftPlus	diag [1,1]	diag [1;8]	2.5	2.5	8
8	3	0.001	SoftPlus	SoftPlus	diag [1,1]	diag [1;8]	2.5	2.5	8
9	3	0.0005	SoftPlus	SoftPlus	diag [1,1]	diag [1;8]	1.5	1.5	8

Figures 8, 9 show the results of deployment using the IA in Case 1. As can be seen from Fig. 8, this IA ensures the deployment of the structure in a given time, but at the same time, V-folding rods rotate with a high angular velocity before they latch, which is undesirable because it increases the loads on the structure. It is possible to reduce these angular velocities by training the IA using Cost 2. Figures 9, 10 demonstrate the results of the structure deployment using such an IA. As can

be seen from Fig. 8–11, in Case 2, the angular velocities and control torque are less than in case 1. In both cases, the deployment is performed in the required time.

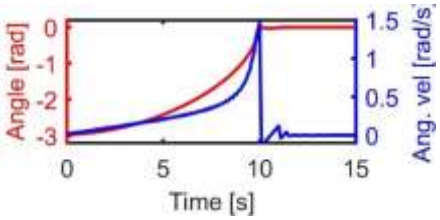


Fig. 8 – State variation in Case 1

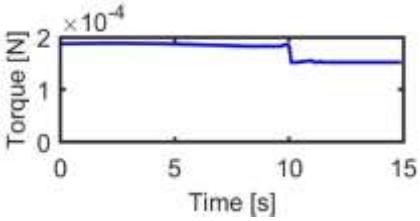


Fig. 9 – Torque variation in Case 1

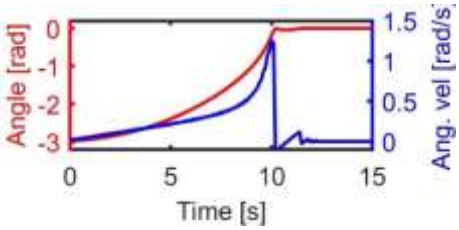


Fig. 10 – State variation in Case 2

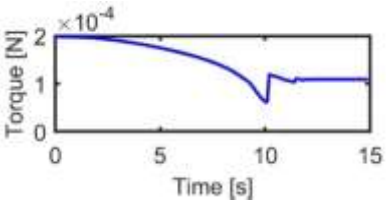


Fig. 11 – Torque variation in Case 2

As can be seen from Fig. 12, 13, when IA is trained using Cost 3, it deploys the structure in the specified time and ensures smoother variations of angular velocities and control torques than when Cost 1 and Cost 2 are used.

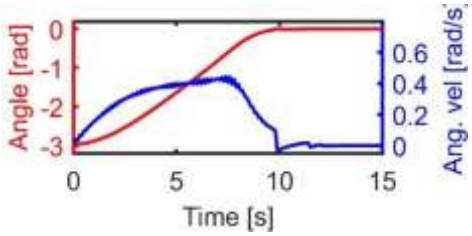


Fig. 12 – State variation in Case 3

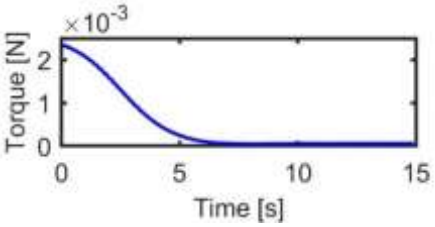


Fig. 13 – Torque variation in Case 3

Case 4 corresponds to the situation when the friction coefficient during deployment testing is lower than during training. It is evident from Fig. 14 that in this case the deployment of the structure occurs significantly faster than required. In addition, at the end of deployment, high angular velocities of the hinge elements are observed.

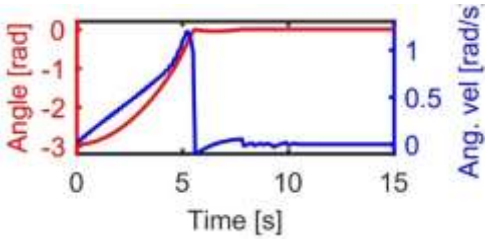


Fig. 14 – State variation in Case 4

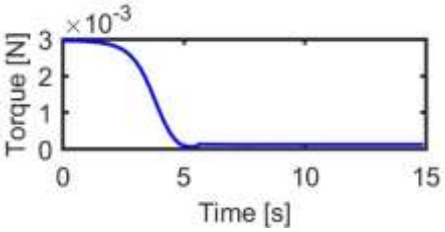


Fig. 15 – Torque variation in Case 4

In case 5, the friction coefficient during deployment testing is greater than during training. From Fig. 16, it is clear that in this case, the deployment of the structure is not completed within the specified time. For such cases, the IA has to be trained additionally. As can be seen from Fig. 18, 19, the IA in both cases

requires about 300 additional training episodes in order to be able to deploy the structure in a given time. In this case the control weights were altered to 0.5.

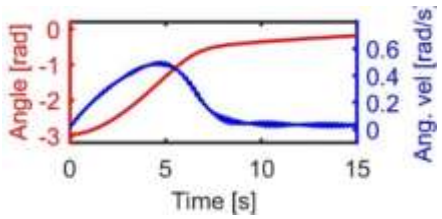


Fig. 16 – State variation in Case 5

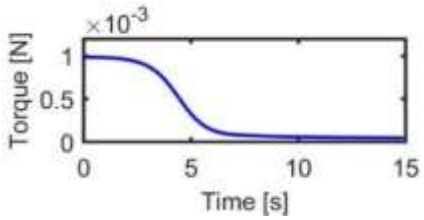


Fig. 17 – Torque variation in Case 5

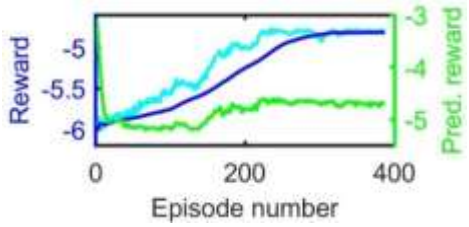


Fig. 18 – Reward variation during additional training in Case 4

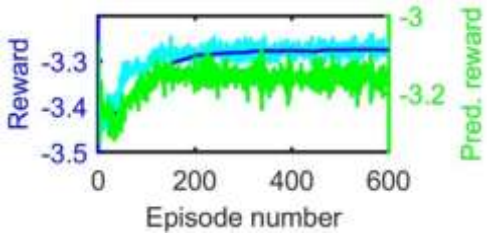


Fig. 19 – Reward variation during additional training in Case 5

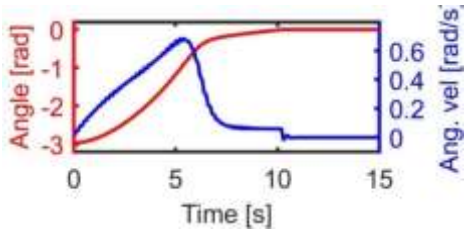


Fig. 20 – State variation for additionaly trained IA in Case 5

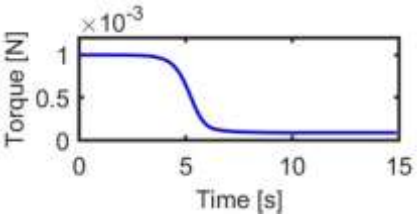


Fig. 21 – Torque variation for additionaly trained IA in Case 5

Figures 22–25 show variation of the comulitive reward during training of the IA using different activation functions at the actor's output. From Figure 22 it is clear that limiting the standard deviation of control actions of the IA in the range of [0...1] ensures a stable training process. At the same time, a greater costrain on the standard deviation of control actions not only slows down the training process, but also makes it less stable. This is clear from Fig. 23, where during training of the IA the standard deviation of control actions is limited to the range of [0...0.1].

Figures 24–25 show cases where the standard deviation of the control actions of the IA is not constrained. In this case, the IA learns faster than in case 3 because it expore state-action space more actively, but its performace may detereorate during trainig.

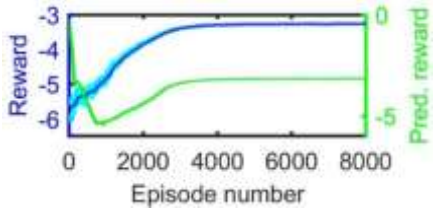


Fig. 22 – Reward variation during training in Case 3

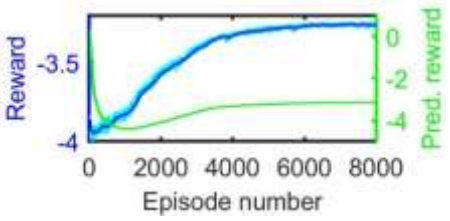


Fig. 23 – Reward variation during training in Case 6

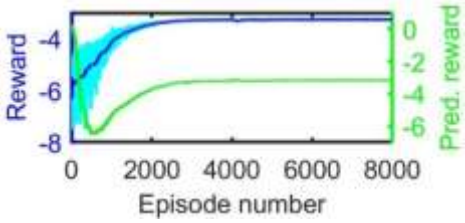


Fig. 24 – Reward variation during training in Case 7

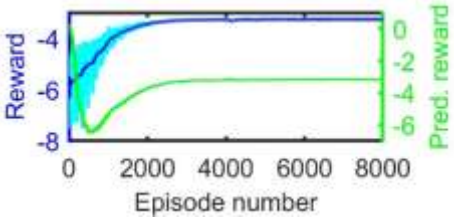


Fig. 25 – Reward variation during training in Case 8

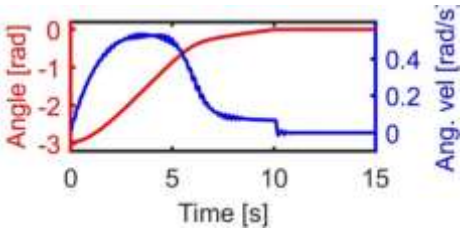


Fig. 26 – State variation in Case 8

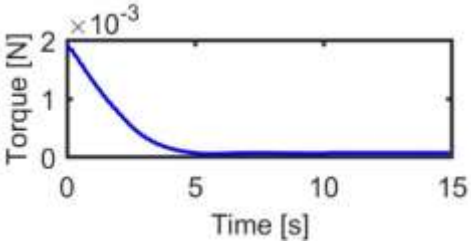


Fig. 27 – Torque variation in Case 8

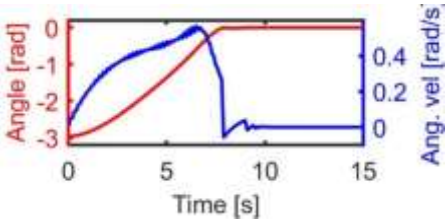


Fig. 28 – State variation in Case 9

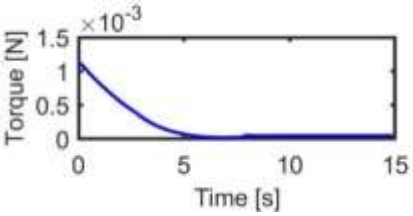


Fig. 29 – Torque variation in Case 9

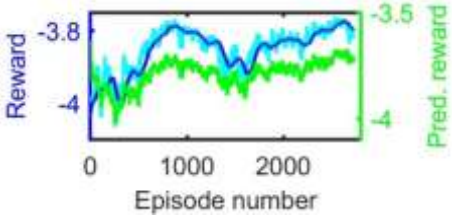


Fig. 30 – Reward variation during additional training in Case 9

Figures 26–30 show the results for the IA, when SoftPlus functions are used at for both mean and the standard deviation output of the NN actor. Such activation functions constrain the output to the region of positive values. It can be seen from Fig. 26 that such an IA provides the deployment of the structure in a given time, while the control torque varies faster from the maximum to the minimum value (Fig. 27). As in the previously considered cases, the application of this IA to a system with less friction leads to the fact that the structure deploys faster than required. An attempt to additionally train the IA In case 9 was not as successful in comparison with the previously considered cases in terms of the requied episodes and stability.

Thus, it can be concluded that tanh activation functions with a subsequent scaling layer are the best option for the actor's output. Such architecture allows for

limiting control actions taking into account the their technical implementation, as well as ensuring fast and stable IA learning.

Conclusion. The article demonstrates the possibility of using the RL methodology to control the deployment of rod transformable structures, which allows overcoming a number of shortcomings inherent in the conventional methodology. In particular, the RL makes it possible to optimize the deployment system using models obtained using specialized software for multibody dynamic simulation, considering the necessary criteria and constrains. In cases where the dynamic properties of the model and the real structure differ significantly, the IA can be fine-tuned. This operation can be implemented through the deployment of the real structure, since the IA requires significantly fewer attempts for the fine-tuning than for the pre-training.

1. *Puig L., Barton A., Rando N.* A review on large deployable structures for astrophysics missions, Acta Astronautica. 2010.Vol. 67, Is. 1–2. Pp.12–26. https://doi.org/10.1016/j.actaastro.2010.02.021
2. *Meguro A., Harada S., Watanabe M*. Key technologies for high-accuracy large mesh antenna reflectors. Acta Astronautica. 2003. Vol. 53. P.899–908. https://doi.org/10.1016/s0094-5765(02)00211-4
3. *Scialino L., Ihle A., Migliorelli M., Gatti N., Datashvili L., Klooster K., Santiago Prowald J.* Large deployable reflectors for telecom and earth observation applications. CEAS Space Journal. 2013. 5. P. 125–146. https://doi.org/10.1007/s12567-013-0044-7
4. *Thomson M.* The AstroMesh deployable reflector. IEEE Antennas Propag. Soc. 2003. 3. https://doi.org/10.1109/aps.1999.838231
5. *Medzmariashvili E., Tserodze S., Sushko A. et al.* Structure, structural features, assembling, and bench testing of the deployable space reflector. CEAS Space J. 2024. https://doi.org/10.1007/s12567-024-00575-7
6. *Rivera A., Stewart A.* Study of Spacecraft Deployables Failures. 19th European Space Mechanisms and Tribology Symposium, Online, September 20–24th, 2021. https://doi.org/10.5281/ZENODO.11425012
7. *Khoroshylov S., Martyniuk S., Sushko O., at al.* Dynamics and attitude control of space-based synthetic aperture radar. Nonlinear Engineering. 2023. Vol. 12 (1). 20220277. https://doi.org/10.1515/nleng-2022-0277
8. *Zhang Y., Duan B., Li T.* A controlled deployment method for flexible deployable space antennas. Acta Astronautica. 2012. Vol.81, Is.1. Pp.19–29. https://doi.org/10.1016/j.actaastro.2012.05.033
9. *Li T.* Deployment analysis and control of deployable space antenna. Aerospace Science and Technology. 2012. Vol. 18, Is. 1. pp.42–47. https://doi.org/10.1016/j.ast.2011.04.001
10. *Zhang Y., Yang D., Li S.* An integrated control and structural design approach for mesh reflector deployable space antennas. Mechatronics. 2016. Vol. 35. Pp.71–81. https://doi.org/10.1016/j.mechatronics.2015.12.009
11. *Zhang Y., Yang D., Sun Z., Li N., Du J.*: Winding strategy of driving cable based on dynamic analysis of deployment for deployable antennas. Journal of mechanical science and technology. 2019. Vol. 33. Pp.5147–5156. https://doi.org/10.1007/s12206-019-0906-9
12. *Peng H., Li F., Kan Z., Liu P.* Symplectic Instantaneous Optimal Control of Deployable Structures Driven by Sliding Cable Actuators. Journal of Guidance, Control, and Dynamics. 2020. Vol. 43. Pp.1114–1128. https://doi.org/10.2514/1.g004872
13. *Goodfellow I., Bengio Y.* A. Deep Learning. Eds. Courville. The MIT press. 2016. ISBN 978-0262035613.
14. *Khoroshylov S. V., Redka M. O.* Deep learning for space guidance, navigation, and control. Space Science and Technology. 2021. Vol. 27, № 6 (133). Pp.38–52. https://doi.org/10.15407/knit2021.06.038
15. *Izzo D., Märtens M., Pan B.* A survey on artificial intelligence trends in spacecraft guidance dynamics and control. Astrodyn. 2019. Vol. 3. Pp.287–299. https://doi.org/10.1007/s42064-018-0053-6
16. *Redka M. O., Khoroshylov S. V.* Determination of the force impact of an ion thruster plume on an orbital object via deep learning. Space Science and Technology. 2022. Vol. 28, № 5 (138). Pp.15–26. https://doi.org/10.15407/knit2022.05.015
17. *Khoroshylov S. V., Wang C.* Spacecraft relative on-off control via reinforcement learning. Space Science and Technology. 2024. Vol. 30, № 2 (147). Pp.3–14. https://doi.org/10.15407/knit2024.02.003
18. *Liu Y., Ma G., Lyu Y., et al.* Neural network-based reinforcement learning control for combined spacecraft attitude tracking maneuvers. Neurocomputing 484. 2022. Pp.67–78. https://doi.org/10.1016/j.neucom.2021.07.099
19. *Gaudet, B., Linares, R., Furfaro, R.* Six degree-of-freedom body-fixed hovering over unmapped asteroids via lidar altimetry and reinforcement meta-learning. Acta Astronaut. 2020. Vol. 172. Pp.90–99. https://doi.org/10.1016/j.actaastro.2020.03.026
20. *Sushko, O., Medzmariashvili, E., Filipenko, L, et al.* Modified design of the deployable mesh reflector antenna for mini satellites. CEAS Space Journal. 2021. Vol. 13. Pp.533–542. https://doi.org/10.1007/s12567-020-00346-0
21. *Khoroshylov S., Martyniuk S., Medzmariashvili E. et al.* Deployment modeling and analysis of mesh antenna consisting of scissor-like and V-folding elements. CEAS Space J. 2024. https://doi.org/10.1007/s12567-024-00584-6
22. *Gerstmayr J., Dorninger A., Eder R. et al.* HOTINT: A Script Language Based Framework for the Simulationof Multibody Dynamics Systems. ASME IDETC/CIE. 2013. V. 7B, V07BT10A047. https://doi.org/10.1115/DETC2013-12299

23. *János Z., Rachholz R., Woernle C.* Field test validation of Flex5, MSC. Adams, alaska/Wind and SIMPACK for load calculations on wind turbines. Wind Energy 19.7. 2016. Pp.1201–1222. https://doi.org/10.1002/we.1892
24. *Lewis F. L., Vrabie D., Syrmos V. L.*, Optimal Control, 3rd Edition. John Wiley & Sons, Inc., New York, USA, 2012. https://doi.org/10.1002/9781118122631
25. *Sutton R. S., Barto A. G.* Reinforcement learning: an introduction. Eds. MIT press, 1998. ISBN 978-0262193986.
26. *Schulman J., Wolski F., Dhariwal P., Radford A., Klimov O.* Proximal policy optimization algorithms. arXiv preprint. 2017. arXiv:1707.06347.
27. *Mnih V., Badia A., Mirza M., Graves A., Lillicrap T., Harley T., Silver D.* Asynchronous Methods for Deep Reinforcement Learning. arXiv preprint. 2016. ArXiv:1602.01783.

Received on March 4, 2025,
in final form on March 24, 2025