Advancing Reinforcement Learning Control: Continuous Scheme and Curriculum Learning for Mechanical Systems

Oleg Rogov[1,2], Peter Manzl[3], Johannes Gerstmayr[3], and Grzegorz Orzechowski[2]

[1]Savonia University of Applied Sciences
[2]Lappeenranta-Lahti University of Technology LUT
[3]University of Innsbruck

May 31, 2024

1 Introduction

Multibody System Dynamics (MSD) is a field of study that concentrates on modeling, simulating, and analyzing the dynamics of systems composed of interconnected rigid and flexible bodies. It is a fundamental approach to understanding and predicting the dynamic behavior of complex mechanical systems through sophisticated mathematical models and computational algorithms. The essence of MSD lies in its capacity to accurately describe the motion, forces, and interactions among the components of a system and with the environment, considering both translational and rotational movements and the constraints that govern these movements [17]. MSD is applicable across various engineering fields, from aerospace and automotive to biomechanics and robotics. This versatility underscores its capability to address the specific dynamics of different systems, enhancing its utility in diverse technological and scientific domains. Also, by accurately predicting system behavior, MSD supports the iterative design and optimization of mechanical systems. Engineers can use MSD to test and refine designs virtually, reducing the need for physical prototypes and accelerating the development process. One important characteristic of MSD is that the detailed models developed using this method make it an excellent foundation for implementing learning-based or model-based control strategies. These strategies, such as Reinforcement Learning (RL) [18] and Model Predictive Control [16], can dynamically adapt to the system's behavior, leading to optimized performance even in complex and changing environments.

Reinforcement Learning, a primary control method used in our research, is a branch of Machine Learning (ML) where an agent learns to make decisions by taking actions in an environment to maximize the cumulative reward under acting according to a certain policy [18]. In RL, rewards may not be immediate. An agent may need to perform a series of actions before receiving a reward, creating a delay between the initial action and the eventual reward. The learning process involves exploring various actions and states (exploration) and exploiting the knowledge gained to make better decisions (exploitation). The exploration and exploitation trade-off is one of the crucial concepts in reinforcement learning (RL), which emphasizes the need for a balance that results in efficient agent training. Achieving this balance is essential for the agent to learn effectively and optimize its performance over time. RL can be divided into two main types: model-free methods and model-based methods. The main difference between them is that in model-free methods the agent

learns directly from interactions with the environment without trying to understand its dynamics, while in model-based methods the agent learns a model of the environment's dynamics and uses this model to plan and make decisions. Model-free methods are generally simpler and more robust in highly dynamic and complex environments, though they require more interactions for an agent to learn effectively. Additionally, RL can be further categorized into traditional Reinforcement Learning (RL) and Deep Reinforcement Learning (DRL). Traditional RL focuses on methods where the agent typically operates with simpler, lower-dimensional state spaces. These methods often rely on tabular representations or linear function approximations. DRL, however, specifically integrates deep neural networks to approximate value functions or policies, so that it can effectively handle high-dimensional state spaces and complex environments. This capability allows DRL to be applied to tasks such as playing video games, robotic control, and autonomous driving [14, 11, 21]. Examples of DRL methods are Advantage-Actor Critic (A2C), Proximal Policy Optimization (PPO) and Deep Q-Network (DQN). A2C is an advanced version of the actor-critic method that uses both policy (actor) and value (critic) functions, with the actor updating the policy based on feedback from the critic, enhancing stability and performance [10]. PPO is a popular policy gradient method known for its robustness and simplicity, utilizing a clipped objective function to ensure stable learning [15]. DQN combines Q-learning [18] with deep neural networks to handle high-dimensional state spaces, approximating the Q-value function, which represents an expected cumulative reward an agent can achieve by taking a specific action in a given state and following the optimal policy, enabling it to learn effective policies in complex environments [11]. In Reinforcement Learning, actions (control inputs) can be either discrete or continuous. Discrete actions are suitable for tasks where the set of possible actions is limited and distinct, such as choosing between different predefined options. Continuous actions, on the other hand, are used in scenarios where actions can take any value within a range, such as adjusting the speed of a car or the angle of a robotic arm.

The application of RL techniques in the context of MSD is still an emerging field, with limited research available. Current studies have primarily focused on traditional control methods, leaving significant potential for further exploration and development of RL-based approaches to enhance the adaptability and efficiency of MSD models. A noticeable study in this area is done by Kurinov et al. [8]. Researchers have developed an autonomous excavator system utilizing RL as a control base integrated with MSD (detailed simulation with hydraulics and sensors). This study was focused on enabling the excavator to perform various tasks autonomously, including excavation and loading. The effectiveness of their approach was demonstrated in achieving autonomous operation with minimal collisions, showcasing the potential for automation in heavy machinery within the construction and mining industries. Additionally, Egli et al. [4] explored the application of reinforcement learning (RL) for adaptive control in hydraulic excavators, aiming to enhance the automation of excavation tasks. Their study introduced a controller capable of adjusting to varying soil properties within a single excavation cycle. Utilizing an RL-based approach, they trained a control policy in simulation, integrating proprioceptive measurements to infer soil characteristics. This method demonstrated significant improvements in adaptability and efficiency compared to traditional excavation techniques. Notably, their controller could operate without explicit soil parameter inputs, relying instead on measurements from hydraulic pressure and kinematic sensors. The experiments conducted on a 12-ton excavator confirmed the controller's ability to maintain high performance across diverse soil conditions, emphasizing the feasibility of RL in real-world excavation scenarios. Furthermore, Osa and Aizawa [13] investigated deep reinforcement learning with adversarial training for automated excavation. They proposed a novel regularization method that employs virtual adversarial samples to mitigate the overestimation of Q-values in a Q-learning algorithm. Their approach, which integrates depth images for excavation planning, demonstrated superior sample efficiency and robustness in policy learning compared to traditional actor-critic methods [18]. This research highlights the

importance of incorporating visual information and advanced regularization techniques to enhance the reliability and effectiveness of RL-based systems.

While RL has shown considerable promise as a control method for complex dynamic systems, it is not without its limitations. One of the primary challenges of RL is its high demand for extensive interactions with the environment to learn effective policies, which can result in long training times and high computational costs. Additionally, RL algorithms often struggle with the exploration-exploitation trade-off, especially in high-dimensional state spaces and non-linear dynamics typical of MSD. These factors can make RL less efficient and robust when applied directly to complex mechanical systems [3]. This is where Curriculum Learning (CL) can provide substantial benefits. CL is about structuring training content and learning experiences in a way that mirrors the progression from simple to complex, akin to how a curriculum in educational settings is designed [1]. This method may not only accelerate the learning process but also enhance the robustness and adaptability of the control strategies. Research by Hacohen et al. [7] delves into the effectiveness of CL in the training of deep neural networks, with a focus on how CL can significantly impact the efficiency of the learning process. Experiments to analyze the impact of curriculum-based learning on the training efficiency of deep neural networks were conducted and the curriculum was structured around a series of tasks, each representing a different level of difficulty, designed to challenge the neural network progressively. This method allowed for the examination of how neural networks adapt their learning strategies in response to escalating task complexity. This work suggests that by implementing a curriculum-based approach deep RL algorithms can develop more robust and effective strategies. The work by Narvekar et al. [12] is particularly instructive for our research as it provides a taxonomy of CL methodologies that can be directly applied to the RL challenges we could face in tandem with MSD. Narvekar et al. categorize CL strategies into three main approaches: task sequencing, transfer learning, and multi-task learning. Task sequencing involves arranging learning tasks in a meaningful order to gradually increase complexity, thereby enhancing the learning efficiency and robustness of RL agents. Transfer learning focuses on leveraging knowledge gained from previous tasks to improve performance on new, related tasks. Multi-task learning allows simultaneous training on multiple tasks to share knowledge and improve overall learning efficiency. In our context, task sequencing is particularly applicable. By structuring the learning experiences from simple to complex tasks, we can systematically improve the RL agent's ability to handle the difficulties of mechanical systems in MSD. This systematic enhancement of learning experiences can mitigate some of the efficiency and robustness issues traditionally associated with RL in dynamic, high-dimensional environments. The study written by Gupta et al. [6] investigates how integrating CL can enhance RL for continuous control tasks, which are highly relevant to MSD. Their research focuses on tasks such as robotic arm manipulation and locomotion, demonstrating that various CL strategies, like progressive task sequencing and parameter adaptation, significantly improve learning efficiency and performance. The process of knowledge transfer within CL can be effectively combined with Transfer Learning (TL) to enhance learning efficiency. Transfer learning focuses on leveraging knowledge gained from previous tasks to improve performance on new, related tasks. As written by Taylor et al. [19], TL emphasizes the importance of prior learning in tackling subsequent, more complex tasks. This approach is directly applicable to the nature of control tasks in MSD, where knowledge from simpler tasks can be transferred to more complex ones, enhancing the overall learning process. Combining CL and TL can thus provide a structured and efficient way to develop robust RL agents for complex mechanical systems. Furthermore, Weinshall's work [20] have delved into the integration of TL within CL, highlighting how accumulated knowledge can significantly boost performance in more complex systems. Their work demonstrates that using a pre-trained neural network on a different task to guide the curriculum can approximate an ideal learning sequence, leading to faster convergence and improved generalization. A paper by Bhati et al. [2] demonstrates the

effectiveness of CL in a multi-agent RL environment. A sequential task structure was implemented where each task increased in complexity and inter-agent dependency. This approach was designed to incrementally develop and refine the cooperative strategies of individual agents within a simulated multi-agent environment, allowing researchers to observe how agents adapted and optimized their performance in progressively challenging scenarios. This finding is crucial for tasks where teamwork among agents is a key factor.

The perspectives of using Curriculum Reinforcement Learning (CRL) in MSD are promising. CRL, which integrates CL with RL, offers a structured approach to tackle the complexities of MSD by gradually increasing the difficulty of control tasks. This methodology not only enhances learning efficiency but also improves the adaptability and robustness of RL agents in managing dynamic mechanical systems. In our prior work by Manzl et al. [9], we evaluated RL algorithms for controlling mechanical systems using discrete control methods, effectively stabilizing single to triple link inverted pendulum systems. The RL approach demonstrated adaptability and effectiveness in managing the dynamics and complexities of these models. Our current study extends this work by applying a novel continuous control method combined with CL technique. Using the same multi-link inverted pendulum on a cart system, we can clearly demonstrate the improvements afforded by these advanced methods. By switching from discrete to continuous action space, we enable smoother and more precise adjustments in system behavior with the faster training time, which are critical when managing the complex dynamics of multibody systems. Moreover, integrating CL accelerates the training process more, allowing RL agents to quickly adapt to increasingly complex tasks. This dual approach not only streamlines control operations but also significantly enhances learning efficiency. Our empirical evidence demonstrates the effectiveness of these strategies in a controlled mechanical environment, highlighting their practical implications. This research paves the way for developing more advanced, autonomous control systems that could transform interactions with mechanical systems across various industries.

2 Methods

2.1 Mechanical model

The system under investigation is a complex multi-link inverted pendulum on a cart. This setup, known for its inherent instability and dynamic nature, is a cornerstone in control theory. The cart, serving as the base platform for the pendulum links, is limited to linear motion along a horizontal track, simplifying the translational dynamics to one-dimensional motion along the x-axis. The cart's movement is controlled by an externally applied force F, which is crucial for the system's stabilization. The magnitude and direction of this force significantly impact the system's dynamics, providing a means to counteract the gravitational torque imposed by the pendulum links.

Attached to the cart is a series of n rigid links, each connected end-to-end by revolute joints. These joints allow for free rotational movement in the vertical plane, and each link i is characterized by length l and angle ϑ. This model, with its high degree of control difficulty and relevance to real-world applications, provides a valuable platform for testing sophisticated control algorithms, including those based on Reinforcement Learning.

2.2 Reinforcement learning with continuous action space

Reinforcement Learning (RL) is a branch of machine learning where an agent learns optimal behavior through systematic interaction with a dynamic environment, aiming to maximize cumulative

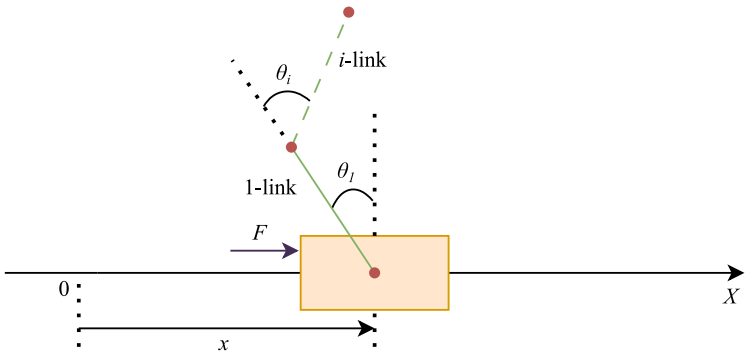


Figure 1: N-link inverted pendulum on a cart representation with link length of 1 m, link mass of 0.1 kg and cart mass of 1 kg [9]

rewards. This learning paradigm is distinct from supervised learning; in RL, the agent is not explicitly instructed which actions to take. Instead, it must explore and discover which actions yield the highest rewards by trial and error, a process often facilitated by a policy—a decision-making function that maps states of the environment to actions to be taken in those states [18].

In our research RL is employed to train agents to manage the dynamics of a multi-link inverted pendulum on a cart, a challenging control problem characterized by the need for precision in dynamic stability. The reward function in RL plays a crucial role as it guides the learning process. For instance, a reward function can be designed to penalize the agent for excessive movement away from a target state or for using too much energy, while rewarding closer approximations of the desired state, such as maintaining the pendulum in an upright position. We use the reward function, which has shown one of the best performances from our previous study

$$r_2 = 1 - w_p \frac{|p_x|}{\chi_{\text{cart}}} - (1 - w_p) \frac{\sum_{j=1}^{\text{N}} |\vartheta_j|}{\text{N}\chi_\vartheta} \tag{1}$$

It combines a tip position p_x with the sum of pendulum link angles $\sum_{j=1}^{N} |\vartheta_j|$, weighted by a factor w_p. χ_{cart} and χ_ϑ are the position and angle thresholds, which differ from 1-link to more link systems. N is the number of pendulum links.

In RL, the choice between discrete and continuous action space might significantly affect the performance of learning algorithms. Discrete action space, used in our previous study, limits the agent's actions to a finite set of possibilities: it either applies a negative or a positive force of the same magnitude to regulate the behavior of the system. In a continuous action space the agent is provided with an interval of the control force

$$F = [-f_{\text{cart}}, +f_{\text{cart}}] \tag{2}$$

From this interval in Eq. 2 the agent is free to choose any suitable control force as an action for learning the stabilization task. Since it provides more possible actions then the discrete action space, it emerges in a faster training of an agent and achieves a more smooth and robust control task execution.

2.3 Curriculum learning implementation

Our approach to Curriculum Learning in the domain of MSD is rooted in the incremental introduction of complexity to the learning environment of the RL agent. Inspired by pedagogical methods where learners progress from simple to complex tasks, we designed a curriculum that starts with the fundamental aspects of mechanical control and systematically advances to more challenging scenarios. We have four parameters in our CL-scheme and their roles are explained in the Table 1. Modelled spring-damper system restriction could be described as Proportional-Derivative (PD)

Table 1: Decay types and their descriptions

Curriculum Learning Parameters	**Description**
Control Values	control parameters representing spring-damper values, which influence the restriction of the system behavior
Decay Steps	time steps when the transition to another set of Control Values occurs
Decay Function	describes the law on how the Control Values will be changed
Decay Factor	sets the speed of Decay Function

control scheme, which with its intuitive appeal and theoretical backing, provides a reliable means to achieve initial stability, that is crucial for the early stages of learning in complex dynamical systems [5]. This phase acted as a confidence-building measure, allowing agent to familiarize himself with the basics of the system's dynamics. Following the establishment of basic control proficiency, the curriculum progressed to a gradual reduction of the control assistance (control values), transitioning responsibility from the control algorithm to the RL agent. This methodical change was governed by predefined decay functions that dictate the rate and pattern of PD assistance withdrawal. The choice of function was determined by the desired learning trajectory and complexity of the task at hand. The comparison of decay functions behavior, which are used in our study, is showed at the Figure 2 In general, decay function could be any function, which will reduce the control values via some mathematical law. The problem is - how to select the best one for our control task? For this we have conducted an analysis of them and the results of it are described in Results section.

3 Results

In this section we examine the performance of the standard RL algorithm PPO using it for a continuous action space in comparison with the discrete, the results of which we took from our previous paper. First subsection presents the results achieved while comparing the trained agent with the same initial parameters using continuous and discrete control schemes. Second subsection describes the work of an agent in a modified environment, where it operates. In the last subsection we present the role of CL in the enhancement of the overall RL agent training.

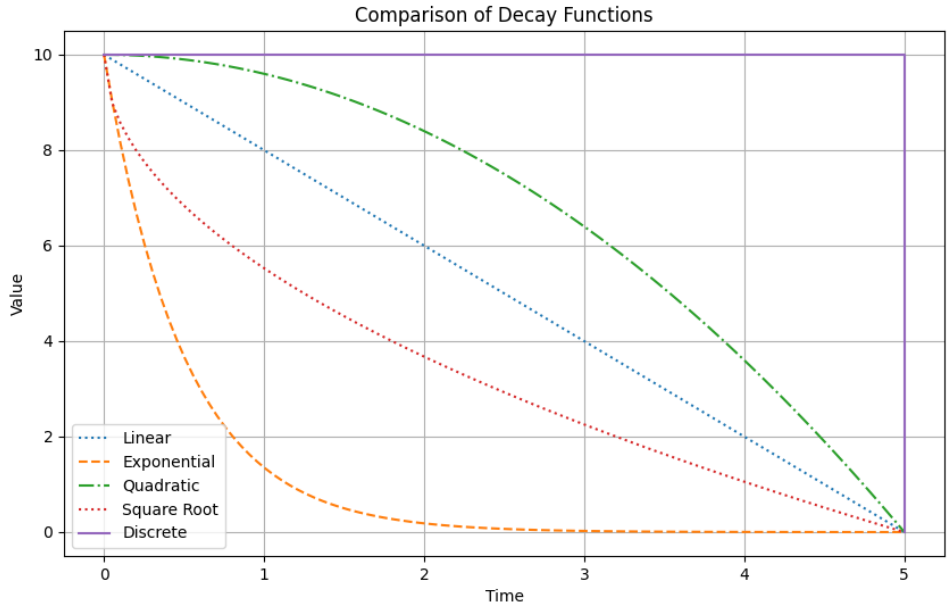


Figure 2: 5 decay functions comparison for a given set of control values [10, 0]. At the decay step equal to 5 all the functions take the next control value of 0.

3.1 RL with continuous action space

For a better evaluation of the continuous control scheme in comparison to the discrete one, we have created a stability zone, the development of which is described in our work of Manzl et al. [9]. From an engineering standpoint, not only are the randomized tests used for evaluation important, but it is also crucial to identify an area where the agent successfully performs the stabilization task for practical, real-world applications. The stability zone is shown in Figure 3. A heatmap displays test results across a grid of parameters, plotting the link's angle on the X-axis and angular velocity on the Y-axis. The continuous control scheme, which replaces the earlier discrete scheme, demonstrates a broader area of optimal performance and smoother transitions at boundary conditions, suggesting improved control capabilities.

The training speed is also significantly different in terms of reaching the required amount of tests for the discrete and continuous control algorithm while it is not clearly seen for the 1- and 2-link systems. Comparison of it is shown at the Fig 4. To clarify what trains faster, the line, where the agent reaches maximum required amount of successful tests is drawn for each system with the shown time step. For all three cases the difference between the discrete and continuous control algorithm is around 45-50% in reaching the successful training, while for the triple pendulum the stable training using discrete control algorithm has be reached after 500 000 time steps.

Our results show, that not even the operating area of the RL agent is wider and smoother, but the agent also achieves a faster training result.

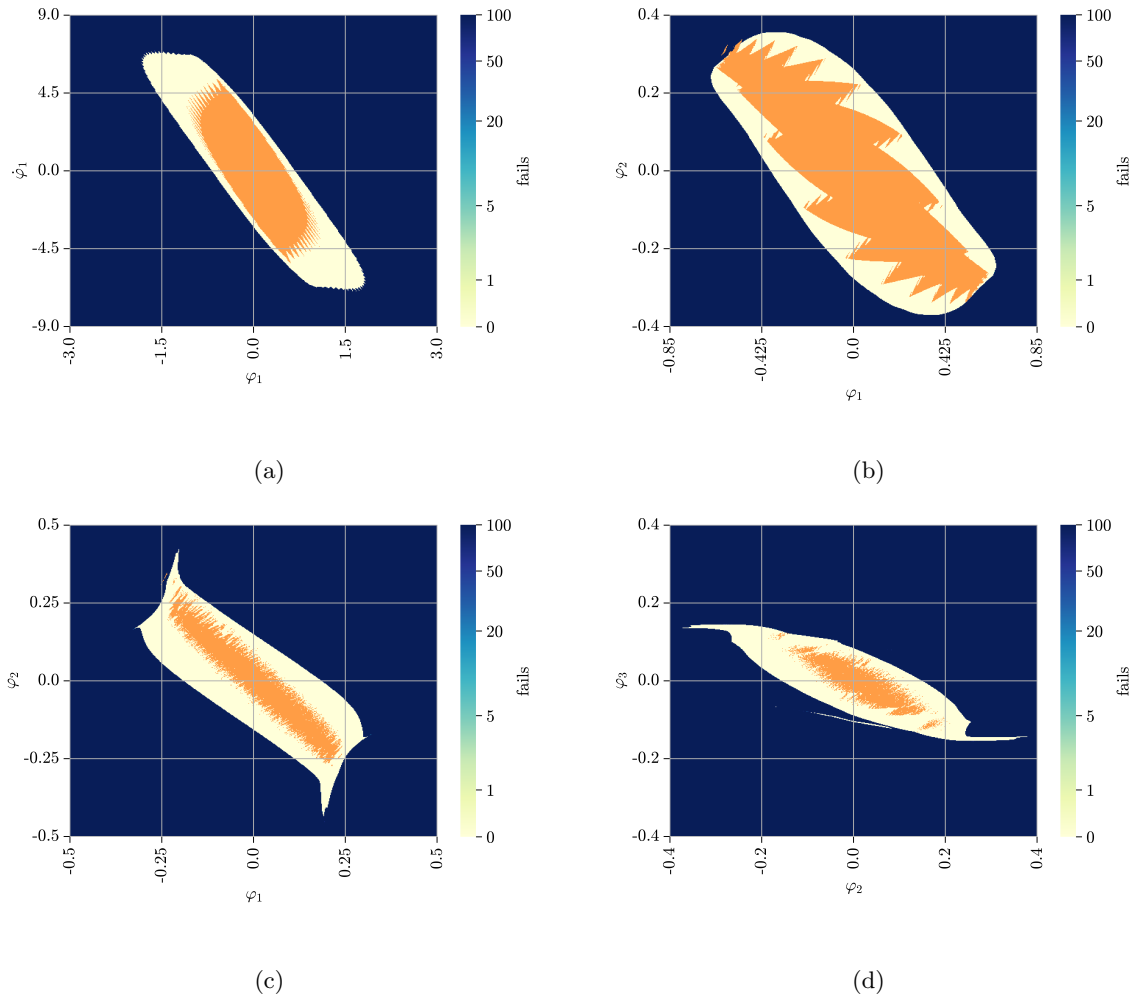
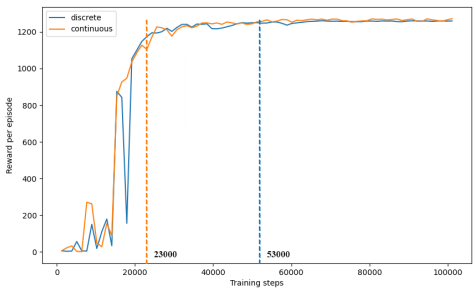


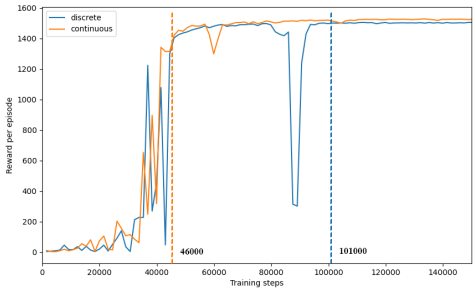
Figure 3: Stability zones comparison of the PPO agent using different control strategies. For the 1-link system (a) the zone axis are link's angle ϕ_1 and angular velocity $\dot{\phi}_1$, while for the 2-link system (b) axis are the pendulum link angles ϕ_1 and ϕ_2. Figures (b) and (c) present the stability zones based on the dependence of pendulum link angles of ϕ_1 and ϕ_2 and of ϕ_2 and ϕ_3. The discrete control stability zone is indicated in orange; continuous is in white beige. Each grid cell represents 100 randomized tests.

3.2 Agent tested on modified environments

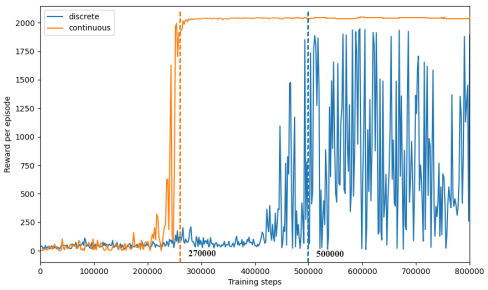
In this section, we examine the impact of changing properties of the environment, namely link length, mass, and added friction, on the stability zones. The investigation is conducted using the inverted double pendulum on a cart model (same continuous agent as in the Figure 3 (b)). In each subplot of Figure 5, the stability zone is determined for the same agent in various modified environments, and the contour of the stability zone from the original environment is overlaid. Friction is implemented using the same method as in our previous research [9].



(a)



(b)



(c)

Figure 4: Training times for the PPO agents for 1-link (a), 2-link (b) and 3-link (c) systems

The observations shows, that increasing the link length will cause the stability zone to expand, while not having the increase of the blind spots, as it was for the discrete control cases [9]. Increasing the mass of the link doesn't provide any differences from the base model, so that we can state that up to the changes of 20% the model is independent from the link mass change. Considering friction the stability zone becomes smaller within each increase of it, but the value of $f_{\text{rel}} = 0.02$ still provides us with the stable zone without causing the agent to fail in the original contour, as it was shown in [9]. It can be concluded that the continuous control scheme is more suitable if friction is involved in the environment model simulation.

3.3 RL training enhancement with CL

To select the parameters needed for the CL implementation the research was conducted in the following fashion. At first, we determine which decay type is specifically the most performing for our particular task of pendulum stabilization. For this we have simulated 150 runs of 5 cases within the control values matrix of the form $\begin{bmatrix} 1 & n \\ 0 & 0 \end{bmatrix}$, where $n = 2, 4, 6, 8, 10$ and decay steps vector $\begin{bmatrix} 0 \\ k \end{bmatrix}$, where $k = 5000, 6000, \ldots, 10000$ to evaluate the performance across different control parameters. The results are presented in a form of a bar plot, which shows the number of agent successes for the whole dataset. It is shown, that the exponential decay type function, provides the best results

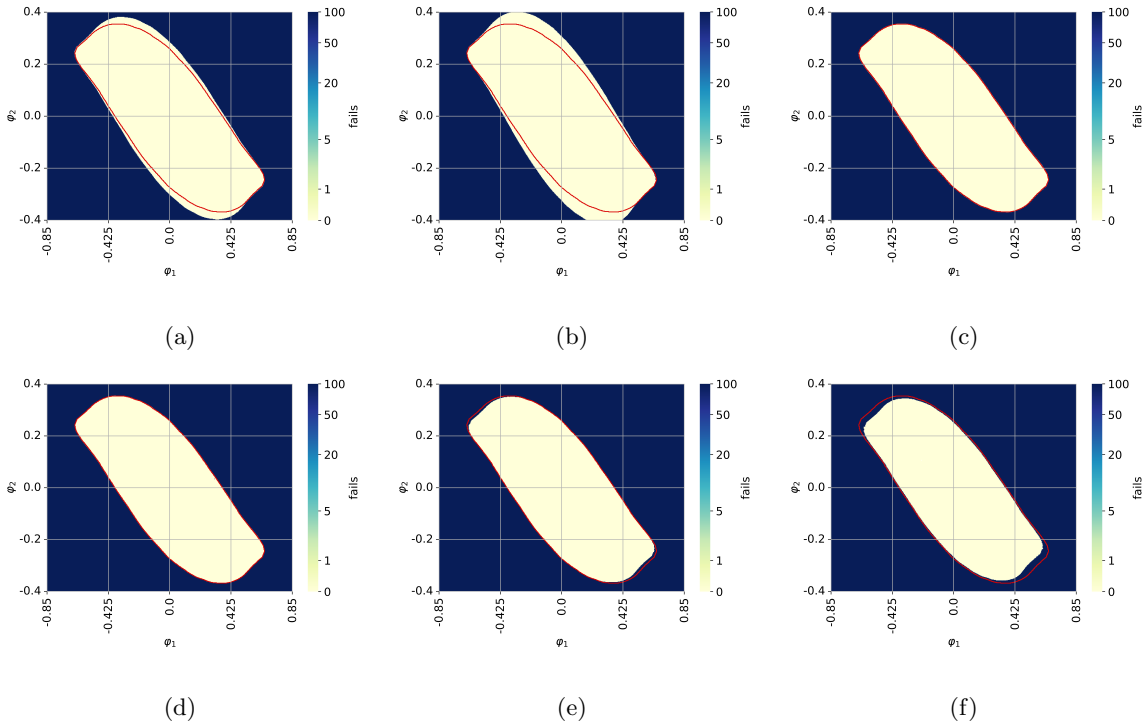


Figure 5: The stability zone for the double link system using the PPO agent, case 0. The red boundary corresponds to that depicted continuous control stability zone in Figure 3 (b). The environment parameters are modified as follows: (a) 1.1 l, (b) 1.2 l, (c) 1.1 m, (d) 1.2 m, (e) $f_rel = 0.01$, and (f) $f_rel = 0.02$.

of having 74% of successes across the dataset. Within this research conducted, in the next steps we have used this decay type to work with for the CL implementation.

4 Conclusions

References

[1] Yoshua Bengio et al. "Curriculum learning". In: *Proceedings of the 26th Annual International Conference on Machine Learning* (2009), pp. 41–48.

[2] Rupali Bhati et al. "Curriculum Learning for Cooperation in Multi-Agent Reinforcement Learning". In: *37th Conference on Neural Information Processing Systems (NeurIPS)* (2023). URL: https://ar5iv.org/html/2312.11768.

[3] Gabriel Dulac-Arnold, Daniel Mankowitz, and Todd Hester. "Challenges of real-world reinforcement learning". In: *arXiv* (2019).

[4] Pascal Egli et al. "Soil-Adaptive Excavation Using Reinforcement Learning". In: *IEEE Robotics and Automation Letters* 7.4 (2022), pp. 1–8. DOI: 10.1109/LRA.2022.3189834.

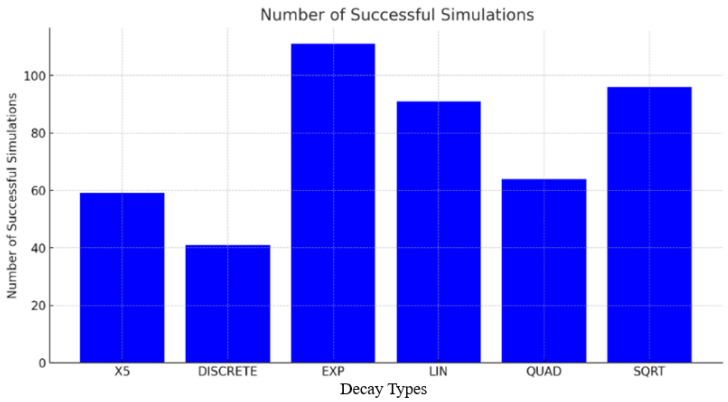


Figure 6

[5] G Franklin, J.D. Powell, and Abbas Emami-Naeini. *Feedback Control Of Dynamic Systems*. Jan. 1994.

[6] Kashish Gupta, Debasmita Mukherjee, and Homayoun Najjaran. "Extending the Capabilities of Reinforcement Learning Through Curriculum: A Review of Methods and Applications". In: *SN Computer Science* 3.28 (2021).

[7] Guy Hacohen and Daphna Weinshall. "On the power of curriculum learning in training deep networks". In: *Proceedings of the 36th International Conference on Machine Learning* (2019), pp. 2535–2544.

[8] Ilya Kurinov et al. "Automated Excavator Based on Reinforcement Learning and Multibody System Dynamics". In: *IEEE Access* 8 (Jan. 2020), pp. 213998–214006. DOI: 10.1109/ACCESS.2020.3040246.

[9] Peter Manzl et al. "Reliability evaluation of reinforcement learning methods for mechanical systems with increasing complexity". In: *Multibody System Dynamics* (2023).

[10] Volodymyr Mnih et al. "Asynchronous methods for deep reinforcement learning". In: *arXiv preprint arXiv:1602.01783* (2016).

[11] Volodymyr Mnih et al. "Playing Atari with Deep Reinforcement Learning". In: *CoRR* abs/1312.5602 (2013). arXiv: 1312.5602. URL: http://arxiv.org/abs/1312.5602.

[12] Sanmit Narvekar et al. "Curriculum Learning for Reinforcement Learning Domains: A Framework and Survey". In: *Journal of Machine Learning Research* 21.181 (2020), pp. 1–50.

[13] Takayuki Osa and Masanori Aizawa. "Deep Reinforcement Learning with Adversarial Training for Automated Excavation using Depth Images". In: *IEEE Access* 10 (2022), pp. 4523–4535. DOI: 10.1109/ACCESS.2022.3140781.

[14] Loris Roveda et al. "Model-Based Reinforcement Learning Variable Impedance Control for Human-Robot Collaboration". In: *Journal of Intelligent and Robotic Systems* (Nov. 2020).

[15] John Schulman et al. "Proximal policy optimization algorithms". In: *Proceedings of the 34th International Conference on Machine Learning-Volume 70*. JMLR. org. 2017, pp. 3191–3199.

[16] Max Schwenzer et al. "Review on model predictive control: an engineering perspective". In: *The International Journal of Advanced Manufacturing Technology* 117 (2021), pp. 1327–1349.

[17] Ahmed A. Shabana. *Dynamics of Multibody Systems*. 4th ed. Cambridge University Press, 2013.

[18] Richard S. Sutton and Andrew G. Barto. *Reinforcement Learning: An Introduction*. English. second edition. Cambridge, Massachusetts: A Bradford Book, 2018. ISBN: 978-0-262-03924-6.

[19] Matthew Taylor and Peter Stone. "Transfer Learning for Reinforcement Learning Domains: A Survey". In: *Journal of Machine Learning Research* 10 (July 2009), pp. 1633–1685. DOI: 10.1145/1577069.1755839.

[20] Daphna Weinshall, Gad Cohen, and Dan Amir. "Curriculum learning by transfer learning: theory and experiments with deep networks". In: *Proceedings of the 35th International Conference on Machine Learning* (2018).

[21] Shenghan Zhu et al. "An optimization method for the inverted pendulum problem based on deep reinforcement learning". In: *Journal of Physics: Conference Series* 2296.1 (2022), p. 012008.