How to Cite:
Salwan, D., & Kant, S. (2022). Comparison of benchmarking algorithms in Prosthesis. *International Journal of Health Sciences*, *6*(S3), 1731–1741. https://doi.org/10.53730/ijhs.v6nS3.5760

Comparison of benchmarking algorithms in Prosthesis

Deepali Salwan
Associate Professor Delhi University New Delhi
Email: deepalisalwan@gmail.com

Shri Kant
Professor, Sharda University Greater Noida, U.P.
Email: shri.kant@sharda.ac.in

***Abstract*---**Due to the remarkable performance of deep reinforcement algorithms in numerous benchmark tests and current environmental setups, Deep Reinforcement Learning has gained a lot of attention in recent years. In such methods, the combination of Deep Learning, which is already well established and strong, and Reinforcement Learning, which is unique, provides amazing results. To enable valuable technical and practical insights into these methods and their results, it is deemed necessary that this paper provides a comprehensive look at these methods and their results, which is concise, accurate, and comparable. Passive prostheses don't fully restore lost functions like robot prostheses do for amputees with transfemoral amputations. There are, however, several issues that must be resolved before an automatic control system could be developed for prosthetic devices. It is a natural tool to use that is based on reinforcement learning (RL). The controlled prosthesis should be able to adapt to different task environments as quickly and easily as possible when it is controlling the prosthesis with the user in the loop. When an environment changes during a runtime, most real-time agents have to re-learn all the rules. In this paper we will be benchmarking the following algorithms: Proximity Policy Optimization (PPO), Deep Deterministic Policy Gradient (DDPG), Trust Region Policy Optimization (TRPO), and ME-TRPO.

***Keywords*---**PPO, DDPG, TRPO, ME-TRPO, reinforcement learning, deep learning, proesthesis.

International Journal of Health Sciences ISSN 2550-6978 E-ISSN 2550-696X © 2022.
***Corresponding author**: Salwan, D.; Email: deepalisalwan@gmail.com*
Manuscript submitted: 18 Nov 2021, Manuscript revised: 09 Feb 2022, Accepted for publication: 27 March 2022

Introduction

There has been an increase in interest in designing prosthetics to improve human movement due to recent advancements in material science and device technology. Nevertheless, designing these devices is a difficult task since the iteration process may be costly and time consuming through a number of different designs. Additionally, the fact that a great deal of variability exists between individuals makes this task even more challenging [1]. Among the reasons we do not know how humans will adapt to their prosthetic devices is that our insight into the way they will interact with prosthetic devices does not go far enough in understanding how a human will adapt to movement. In this field of biomechanics, physics-based biomechanical simulations are well poised to make significant advancements, as they can be used to perform several different experiments for a very low cost. We have been developing new techniques for training realistic, biomechanical models by means of reinforcement learning techniques [3]. We believe this recent development will increase understanding of human prosthesis interaction between the prosthesis and the body, thus accelerating the development of this field.

Statement of the problem

A healthy leg is of great significance to many people since it improves mobility. It helps them manage daily activities that are part of their daily lives, and enables them to stay independent as they age. Customizing artificial limbs for one individual is a costly and time-consuming process ($50,000 USD). It is complex to design intelligence prosthetics to work with the wide range of differences between individuals (like a person's height, weight, and walking style), since many people have such wide ranges of response to stimulation. There are numerous reasons for this, but one reason is that we do not yet have a complete understanding of how humans interact with prosthetic devices, which makes it difficult to predict how a person will adapt to the device when it is introduced to an environment [2],[5]. In this field, physics-based simulations, which are based on biomechanical principles, are well positioned to advance it since they allow for reduced costs for a wide array of experiments.

Batch policies

- Target Policy pi(a|s): A policy a learning agent is attempting to learn - for example, it is trying to determine the value of the policy as it learns this value function.
- Behavior Policy (a|s): A policy that an agent follows during its interaction with the environment, in other words it follows it for action selection.

On-policy learning

'On-Policy' learning algorithms improve the same policy used when selecting actions. These algorithms have been defined as the ones which evaluate and improve the same policy. Taking that into account, we are going to choose options that are identical to those the agent uses as a basis for action selection [7]. The short explanation is this: [Target Policy == Behavior Policy]. On-Policy algorithms

are some examples of these types of algorithms, including Value Iteration, Policy Iteration, Sarsa, etc.

Off-policy learning

The off-policy learning algorithm measures a policy that is different from a policy the action selection algorithm uses for evaluating and improving actions [5-8]. As a result, [Target Policy != Behavior Policy]. There are several off-policy learning algorithms from which to choose, such as Q-learning, expected sarsa (which can act either way), etc.

Proximal policy optimization

By using a natural policy gradient, policy gradient methods solve the convergence problem of policy gradient methods. Although natural policy gradient is scalable for small problems, in practice, it involves a second-order derivative matrix. Real-world tasks require a high level of computational complexity. A second-order method is approximated by intensive research to reduce the complexity [3]. An alternative approach is used by PPOs. As opposed to imposing a rigid requirement, the objective function formalizes the constraint as a penalty. The Gradient Descent method can be used to optimize the objective by not avoiding the constraint at all costs. By adding PPO, a soft constraint to the problem that is an initial point of optimization can be implemented. Although we may occasionally make a mistake when optimizing our designs, we still are able to accomplish the task quickly without sacrificing quality. This balance of performance with simplicity has been shown by experimental results to have the best performance.

Deep deterministic policy gradient

There is a model-free algorithm called Deep Deterministic Policy Gradient (DDPG) that is used to learn continuous actions without setting up any model. Deep Q-Network (DQN) and Deterministic Policy Gradient (DPG) are combined. This system is based on DPG, which can operate over continuous action spaces, and it makes use of Experience Replay and slow-learning target networks from DQN [8].

Trust region policy optimization

The TRPO algorithm maximizes policies using gradient descent in reinforcement learning. Algorithm-free methods like policy gradients don't rely on an environment model and are more stable in practice. Because of this, they tend to be difficult to apply to large, non-linear problems even though they are straightforward to solve. By combining insights from optimization theory and reinforcement learning [12], TRPO develops an algorithm that ensures monotonic improvement within certain constraints. Almost all new algorithms begin with this algorithm.

Model-ensemble trust region policy optimization

The success of model-free reinforcement learning (RL) methods is heightened by recent advances in deep learning, which have made it possible for these methods to succeed in a growing number of tasks. In spite of these, they are not very useful in real-world applications due to the high complexity of sample data, which interferes with their use [15]. On the other hand, model-based reinforcement learning offers promising results but requires careful tuning to achieve its potential, and to date it has been mainly successful in domains where learning is restricted to simple models and modeling effort is nominal.

Nature of environment

OpenSim ProstheticsEnv enables us to model one human leg and implant prosthetics into another leg using a computer environment. An open-source human model simulator called OpenSim allows users to see joint rotations, muscles, tendons, and various kinds of actions. The reward is measured as the distance between a previous velocity and the desired velocity [7].

$$R_t = 9 - (3 - V_t)^2$$

Hroizontal velocity represents the "V_t"

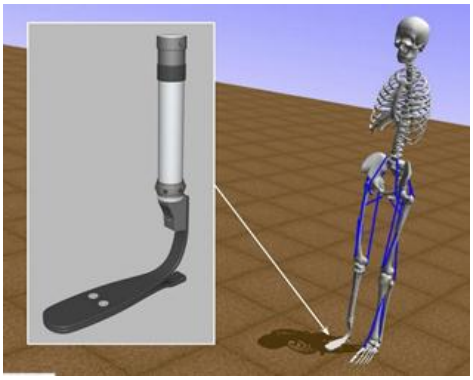

Figure 1. Proesthetic OpenSim Environment

Algorithms of deep learning (RL)

The addition of reinforcement learning (RL) will make it easier for prosthetic devices to adjust to changes in walkers and differences in walking environments that may differ between individuals. This is a novel paradigm of machine learning which can discover the optimal policy to follow in order to achieve a sequential decision-making task without having to know the environment completely [6]. For maximizing the reward Rt, the agent will have to investigate the environment by taking actions and change the policy in accordance with the reward function. Our training method involves the follow four components: PPO, DDPG, ME-TRPO, and TRPO. DDPG uses deep function approximators rather than models or models controlled by actors to the convolution of policy spaces that are high-dimensional, continuously structured, and model-free.

Proesthesis based on 3D models

With the advent of 3D printing technology, upper limb prostheses are becoming an integral part of the technology in the hopes of providing timely access to those who require them. In the clinical setting rapid production of models has proven to be effective, however human health concerns are commonly raised about the safety of 3D-printed parts in everyday use. In recent years, many organizations have developed body-powered 3D printed prosthetic hands, including e-NABLE, Robohand, and Limbitless Solutions [9]. Even though the e-NABLE hand has the capability to perform multiple tasks, it has difficulty performing others because the fingers are designed in such a way that they bend together, causing each finger to move in line with the rest. Because of this, each finger struggles to pick up the object. In addition, many 3D printed prosthetics are not able to withstand large loads. This is because they are made from fragile materials that break down easily.

Epsilon greedy

Exploration gives an agent the opportunity to better understand each action, and therefore enhance its overall performance. In the future, decisions can be made more accurately by improving the accuracy of the estimated action-values. In contrast, Exploitation, like greed, takes advantage of a current action-value estimate to choose the most lucrative action [17]. Nevertheless, being greedy can lead to sub-optimal behavior if you equate action value estimates with expected rewards. Exploration helps agents estimate their action-values more accurately. Exploitation could result in a higher reward. The dilemma of exploring and exploiting at the same time, also known as the exploration-exploitation dilemma, cannot be solved. Exploration and exploitation are balanced using Epsilon-Greedy by choosing between those two separately at random. Exploitation by the epsilon-greedy is most common when there is a small chance of exploration. Epsilon is the probability of exploring [19].

Markov-decision process

A form of Machine Learning that entails reinforcement learning is called reinforcement learning. The goal of this technique is to enable a machine or software agent to determine the optimal behavior within a particular situation to maximize its performance. The agent needs a simple reinforcement signal, which is similar to a reward-based stimulus, so it can learn how to act appropriately [7]. This issue can be tackled in a number of different ways, each of which focuses on a different aspect of it. Essentially, Reinforcement Learning is a sort of algorithm that is based on a type of problem, and each of the solutions to that problem is considered to be an algorithm for Reinforcement Learning. As a result of the problem, an agent is required to determine based on his current state what would be the best course of action [10]. Markov Decision Process is the name of the problem that is referred to when repeatedly repeating this step. Models of Markov decision processes (MDPs) include:

- World states S are possible states of the world.
- There are a number of Models

- Actions that may be taken A
- It is a function of reward value R(s,a)
- This is a policy-driven solution for Markov Decision Processes

State

An agent's "state" can be represented by a set of tokens known as a state.

Model

Various types of Models (sometimes called Transition Models), describe how actions affect states. The transition T(S, a, S') occurs when we are in state S, and we take action 'a' (S and S' may be the same). The stochastic probability of reaching a state S' is P(S',S,a), which is used for stochastic actions (noisy, non-deterministic) [10][12]. In addition to the Markov property, actions in a state will only affect that state, regardless of its past history.

Action

All possible actions are included in Action A. When in state S, A(s) describes what can be done.

Reward

There are several types of rewards. If you are in the state S, you will receive R(s). (S,a) indicates the amount of reward you receive for taking action in a certain state [12]. If you are in state S, take action 'a' and end up in state S', then R(S,a,S') tells you the reward you will receive.

Policy

The Markov Decision Process is solved by a Policy. S is mapped to a by a policy. While in state S, the action 'a' must be performed [10].

Methods

A RL agent is tuned for a new user and then tuned so that it is efficient in terms of training time and sample output. We consider the possibility of building a representation of potentially transferrable knowledge across subjects in order to take advantage of previous knowledge and information. This study considers eliciting the knowledge that might be available from ablebodied (AB) subjects for the future design of rational sequential control algorithms for amputees and AB subjects, respectively [11]. A type of transfer learning which has been gaining a lot of attention in recent years is transfer learning, where it depends on storing knowledge gained from solving a problem, and be applied to solving another problem which seems similar to the one being solved. As stated in the literature, general transfer learning involves learning the relevant parameters for prosthetics in order to use that knowledge to tune parameters for amputee subjects [11],[13].
A value function representation decouples reward dynamics from environment dynamics in a world where rewards change, but environments stay the same.

This is how [17] addressed this problem. The individual who proposed the learning architecture provided a framework in which control knowledge is embodied in behavioral skills and a representation hierarchy, separating subgoals to allow learning to take place on a smaller state space. It was proposed in [20] that apprentices learn target skills for implementation across domains, such as balancing a cart-pole and balancing a bike, for example.

System of proestesis of wearer

An angle sensor was also installed at the rotating knee joint of the robotic knee prosthesis used in this study, as well as a load cell in the pylon to determine the force applied by the pylon. Impedance controllers used commonly in finite-state machines are used to control the robotic knee prosthesis. Cycles of stance flexion, stance extension, swing flexion, and swing extension were assigned to the four finite states corresponding to one gait cycle [14].

$$T_m = k_m(\theta - \theta_{cm}) + b_m\omega$$

Swing flexion

During automatic adaptation for swing flexion, the maximum angle of flexion for the swing has been limited. An anatomically correct prosthetic knee was a nonlinear system in its wearer and the prosthetic knee joint. By simply introducing fuzzy logic or fuzzy inference to the nonlinear system, it was very easy to get good control of it [18].

Experiments

Experiments are conducted as follows:

- OpenSim ProstheticsEnv, 1500 timesteps and 2018 episodes will give an agent more time to walk or stand up when running RL algorithms (ME-TRPO, PPO, DDPG, and TRPO).
- In this instance we modify DAgger to consider the timestep reward of the expert agent and the target agent for labeling the target agents action. The expert keeps the target agents action and the target keeps their action for the timestep: the expert agent retains the target agents action if it is less than the target agent's reward.
- The goal of this training was to get the DDPG agent to perform a standing up task (around +100) in a positive manner.
- Our evaluation of the DAgger algorithm uses that agent as a source of expertise.
- With the epsilon-greedy method, the expert action algorithm has the option to select between taking the target action with a probability of s or taking the with a probability of s.
- In the comparison, we use the target action value instead of the timestep reward, and we calculate the episode reward once all timestep rewards have been accumulated from states and actions.

Protocol of experiments

A treadmill with an experimental knee prosthesis and an IRB-approved and signed consent was set up for six subjects to walk on. Reinforcement learning tuned the control parameters while they walked. Prosthetists fitting powered knee prostheses to amputees for the experimental study fitted the amputee subjects with a powered socket that was adapted to their daily life. A powered knee prosthesis was worn by subjects in able-bodied condition with an L-shaped leg adaptor to allow them to walk normally [19]. For more than six sessions, all subjects walked with the experimental 0.8 m/s speed on a treadmill with a prosthesis until it became natural for them to walk without holding the handrail.
Here is the calculation method that was implemented as part of the impulse calculation described above. Depending on the wearer's ground reaction force and stance phase, three phases were defined: 2) dual leg support (DLS), 2) one-limb support (SS), 3) twin-leg support (TDS). Therefore, by calculating the impulses generated in a gait cycle during each period of time, we were able to perform an impulse analysis [17-20]. It was found that the ground reaction force was separated into two components when performing single limb stances: the positive component as a braking impulse and the negative component as a propelling impulse. In order to calculate the brake impulse, it was necessary to integrate the ground reactions force of the initial upright position during the double-limb grasping phase as well as the negative reaction force of the initial whole limb stance. An automatic method was used to calculate the propulsive impulse by integration. When a terminal double limb supported stance is taken, the ground reaction force is the negative component of that force, while when a single limb supported stance is taken, it's the positive component.

Evaluation and Results

In contrast with TRPO, PPO, and DDPG, ME-TRPO achieved maximum reward mean (These are depicted in Table 1 and Figure 2); however, it takes a longer time to do so, as it must find the inverse of a matrix, which is time-consuming. It is true that the agent, despite this reward, cannot walk for longer than one step, and sometimes, it falls before he can even walk one step. Based on the comparison of Dagger algorithm and other algorithms that balance exploiting and exploring (This is seen in Figure 2), Dagger algorithm achieved the best average reward. These are the following reasons we discussed.

- A high reward can be obtained through expert policy
- Experts and naive agents are very similar
- Increasing the number of iterations will take more time for the naive agent to run

Table 1
Algorithms comparison

Algorithm	Maximum Reward	Mean Reward
PPO	78	-64
DDPG	105	-48
TRPO	223	48
ME-TRPO	276	67

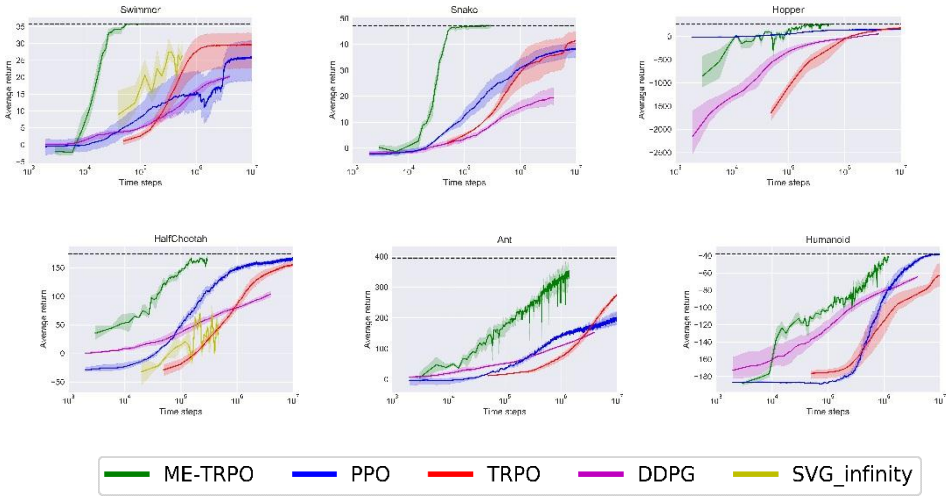

Figure 2. Plotted Graph

Timestep reward modification has the main problem that it compares rewards during different periods of time (short-term) while being aware of the large variation between them. Action-value can be small when there is little variation between episodes. In this case, the naive agent has received a greater reward than the expert agent, so the agent could change roles; instead of being an expert, the agent could become naive, which will significantly reduce the training time for both.

Conclusion

Using an innovative application of RL-based prosthesis tuning, which allows evaluation of different control parameter combinations for the prosthetic knee, we have shown that the wearer's spatiotemporal gait is sensitive to the mechanics of the knee prosthetic. By adjusting only the prosthesis mechanics on both limbs, changes were elicited in stance length as well as time-taken; an increased stance time generally induced a decrease in step length on both limbs, which in turn altered the symmetry indexes significantly. There were likely to be passive or involuntary changes observed movement of the human limb intact. We did not tell the subjects when the prosthesis was adjusted or instructed the subjects to alter their gait, it is more than likely that these changes occurred passively. Several subjects with an appropriate prosthesis control were able to achieve symmetrical step lengths, the control prostheses we tested on our subjects had no stance symmetry or time symmetry whatsoever, even though none of the controls we tested had. It turned out that prosthesis control can be optimized to maximize temporal-spatial symmetry in normal gait, however, only maximizing normative symmetry while adjusting control can produce a desirable gait. A robot knee prosthesis fitted with fixed control and mechanics could still improve gait symmetry when enhanced by human active adaptation. The wearer-prosthesis system could further improve its gait symmetry by combining rehabilitation training with prosthesis optimization.

As a means to accelerate the learning process in a humanoid environment, imitation learning has been applied. The naive agent achieves convergence in five iterations, whereas for the expert, it takes 100 iterations. This means that there is a reduction of 95% in training time. DAgger found a balance between exploring and exploiting in both the exploitation and the exploration phases of the algorithm and achieved the optimal average reward. Our algorithms will be more effective if there is at least some variation between expert and naive agents and in the future, we plan to do this by applying the method of imitation learning to prosthetic legs by applying human leg learning to normal human legs. It would be important to compare them in terms of their differences and similarities in order to be able to figure out how they differ.

References

1. Attia, Alexandre & Dayan, Sharone. (2018). Global overview of Imitation Learning.
2. Cheng, Qiao & Wang, Xiangke & Shen, Lincheng. (2017). An Au- tonomous Inter-task Mapping Learning Method via Artificial Neural Network for Transfer Learning. 10.1109/ROBIO.2017.8324510.
3. Farina, D., Jiang, N., Rehbaum, H., Holobar, A., Graimann, B., Dietl, H., et al. (2014). The extraction of neural information from the surface EMG for the control of upper-limb prostheses: emerging avenues and challenges. IEEE Trans. Neural Syst. Rehabil. Eng. 22, 797–809. doi: 10.1109/tnsre.2014.2305111
4. Feix, T., Pawlik, R., Schmiedmayer, H.-B., Romero, J., Kragic, D., and Kragi, D. (2009). "A comprehensive grasp taxonomy," in Robotics, Science and Systems Conference: Workshop on Understanding the Human Hand for Advancing Robotic Manipulation, Poster Presentation (Seattle, WA), 2–3.
5. Fougner, A., Stavdahl, O., Kyberd, P. J., Losier, Y. G., and Parker, P. A. (2012). Control of upper limb prostheses: terminology and proportional myoelectric control—a review. IEEE Trans. Neural Syst. Rehabil. Eng. 20, 663–677. doi: 10. 1109/tnsre.2012.2196711
6. Fukuda, O., Tsuji, T., Kaneko, M., Otsuka, A., and Tsuji, O. F. T. (2003). A human- assisting manipulator teleoperated by EMG signals and arm motions. IEEE Trans. Robot. Autom. 19, 210–222. doi: 10.1109/tra.2003.808873
7. Gijsberts, A., Atzori, M., Castellini, C., Muller, H., and Caputo, B. (2014). The movement error rate for evaluation of machine learning methods for sEMG-based hand movement classification. IEEE Trans. Neural Syst. Rehabil. Eng. 22, 735–744. doi: 10.1109/tnsre.2014.2303394
8. Goodfellow, I., Bengio, Y., and Courville, A. (2016). Deep Learning (Book in preparation). Cambridge, MA: MIT Press.
9. Hargrove, L., Losier, Y., Lock, B., Englehart, K., and Hudgins, B. (2007). A real-time pattern recognition based myoelectric control usability study implemented in a virtual environment. Conf. Proc. IEEE Eng. Med. Biol. Soc. 2007, 4842–4845. doi: 10.1109/IEMBS.2007.4353424
10. He, K., Zhang, X., Ren, S., and Sun, J. (2015). "Delving deep into rectifiers: surpassing human-level performance on imagenet classification," in IEEE International Conference on Computer Vision (ICCV 2015) (Santiago, CL), 1026–1034.

11. Hinton, G., Deng, L., Yu, D., Dahl, G., Mohamed, A., Jaitly, N., et al. (2012). Deep neural networks for acoustic modeling in speech recognition: the shared views of four research groups. IEEE Signal Process. Mag. 29, 82–97. doi: 10.1109/msp. 2012.2205597
12. Jiang, N., Tian, L., Fang, P., Dai, Y., and Li, G. (2013). Motion recognition for simultaneous control of multifunctional transradial prostheses. Conf. Proc. IEEE Eng. Med. Biol. Soc. 2013, 1603–1606. doi: 10.1109/EMBC.2013. 6609822
13. Jiang, N., Vujaklija, I., Rehbaum, H., Graimann, B., and Farina, D. (2014). Is accurate mapping of EMG signals on kinematics needed for precise online myoelectric control? IEEE Trans. Neural Syst. Rehabil. Eng. 22, 549–558. doi: 10.1109/TNSRE.2013.2287383
14. J.J. Zhu, DAgger algorithm implementation, (2017), GitHub repository
15. Joshi, Girish & Chowdhary, Girish. (2018). Cross-Domain Transfer in Reinforcement Learning using Target Apprentice.
16. Kamakura, N., Matsuo, M., Ishii, H., Mitsuboshi, F., and Miura, Y. (1980). Patterns of static prehension in normal hands. Am. J. Occup. Ther. 34, 437–445. doi: 10. 5014/ajot.34.7.437
17. Kato, R., Yokoi, H., and Arai, T. (2006). "Competitive learning method for robust EMG-to-motion classifier," in Proceedings Intelligent Autonomus Systems (Tokyo), 946–953.
18. Kõiva, R., Hilsenbeck, B., and Castellini, C. (2012). FFLS: an accurate linear device for measuring synergistic finger contractions. Conf. Proc. IEEE Eng. Med. Biol. Soc. 2012, 531–534. doi: 10.1109/EMBC.2012.6345985
19. Lillicrap, Timothy & J. Hunt, Jonathan & Pritzel, Alexander & Heess, Nicolas & Erez, Tom & Tassa, Yuval & Silver, David & Wierstra, Daan. (2015). Continuous control with deep reinforcement learning. CoRR.
20. T. Garikayi, D. van den Heever and S. Matope, (2016), Robotic prosthetic challenges for clinical applications, IEEE International Conference on Control and Robotics Engineering (ICCRE), Singapore, 2016, pp. 1-5. doi: 10.1109/ICCRE.2016.7476146
21. Zardoshti-Kermani, M., Wheeler, B. C., Badie, K., and Hashemi, R. M. (1995). EMG feature evaluation for movement control of upper extremity prostheses. IEEE Trans. Rehabil. Eng. 3, 324–333. doi: 10.1109/86.481972