Association Journal of Interdisciplinary Technics in Engineering Mechanics, Vol. 3, No. 3, 2025
ISSN: 3049-0898

Joint Parameter Calculation in Robotic Arms Using the Denavit-Hartenberg Transformation Algorithm

Dr. Leonardo Costa[1], and Dr. Junpei Takahara[2]

[1]Federativa College of Technics, Brazil.

[2]Federativa College of Technics, Brazil.

Abstract--- Getting the joint angles and lengths just right is a must if robotic arms are going to move smoothly and hit their marks, especially in busy factories or delicate hospital rooms. In this paper, we rely on the Denavit-Hartenberg (DH) method, a clear step-by-step way to show how each arm segment sits in relation to the next, for our calculations. Instead of wrestling with every curve and corner, the DH scheme shrinks the job to four numbers for each pair of links: how long the link is, how much it twists, where the joint pivots, and how far the link starts from that pivot. By stacking these numbers into tidy transformation tables, we can reliably work out where a multi-jointed arm ends up most of the time. Here, we run the standard DH process on a mix of robot setups, from a simple three-joint arm to a whole six-joint rig. Our method walks through forming each matrix, adding them up, and finally showing the tool's tip sits in space relative to the base. When we compare the numbers that come out of the simulation with what we recorded in the real world, they line up closely, proving that DH still delivers the precision many modern jobs demand. The system now works hand-in-hand with inverse-kinematics tools so that planners can steer a robot in real time. Engineers kicked the code around in MATLAB and Python, and both show that the setup moves easily from one platform to another. In short, the Denavit-Hartenberg rules still provide a fast, solid way to figure joint angles and study motion, opening the door to more brilliant, more flexible robot brains. Looking ahead, the team plans to stretch the model so it also fits robots with extra joints or joints that bend in tricky ways.

Keywords--- Joint Angle, Robotic Arm, Kinematics, Movement Steps, DH Parameters, Angular Motion, Actuator Control, Serial Manipulator, Motion Analysis, Pose Estimation

---

Received: 12 - 06 - 2025; Revised: 07 - 07 - 2025; Accepted: 18 - 08 - 2025; Published: 30 - 09 - 2025

I. Introduction

Robotic arms now sit at the heart of many factories, showing up in car plants, airplane shops, hospitals, farms, and busy shipping centers around the world. Built to swing, twist, and grip just like a human arm, these machines handle chores such as welding, painting, stacking boxes, or even steering a delicate laser during surgery. In high-stakes settings like smartphone assembly lines or brain-operation theatres, they outlast tired workers, follow the same path each time, and never shake, making workplaces faster and safer (Craig, 2009). And with the rise of innovative Industry 4.0 factories, adding cameras and artificial-intelligence software has turned many of these arms into surroundings-aware helpers that decide on the fly (Zhang & Rodriguez, 2023). For any robotic arm to do its job well, engineers must nail the math that describes every joint angle and drive belt. Those numbers decide where the gripper points in relation to the body, a detail that guides route planning, speed controls, and whether the job gets finished without a hitch. Mistakes in that math can cause a tip to wobble, two parts to crash together, or a high-DOF robot to drift far off track, and minor errors ripple up to big problems when many joints are in play (Denavit & Hartenberg, 1955). So, solid joint models matter for real machines on the shop floor and also for online testers, digital twins, and students learning the craft in virtual classrooms. The Denavit-Hartenberg (DH) system gives engineers a tidy playbook for figuring out how robotic arms move. Instead of wrestling with bulky formulas for every single joint, it squeezes each link-and-joint pair into just four numbers: link length, twist, angle, and offset. By stacking these pieces into simple matrices, the DH route lets the arm's tip friends out to see where they are, step by step, starting from the base. Because the trick is clear, quick to calculate, and works with simulation tools and inverse kinematics alike, the method snagged research attention right after Denavit and Hartenberg published in 1955 (Petrova & Kowalski, 2025).

Newer ideas like the modified DH convention keep the core spirit while fine-tuning details for different robot shapes, so the model still sits at the heart of labs and factories worldwide today (Lee et al., 2015).

II. Literature Review

Most past work on figuring out a robot arm's joint angles has tried out both step-by-step math and computer number-crunching, especially now that today's robots have so many moving parts. Early papers treated the arm as a set of stiff bars and used simple drawings to guess where each piece should point and how high it should sit (Aswathy, 2024). Once newer designs packed in extra joints and twisted links, engineers leaned on stricter math like matrices and trigonometry so no detail slipped through the cracks. Modeling programs and powerful simulators then let teams build virtual arms, play with different setups, and check whether their equations matched what the screen showed. Through all this, keeping the models exact, quick to solve, and able to grow with the project has been the steady goal, while a fresh push now asks how these calculations can run on the fly inside real-time control loops (Siciliano et al., 2009). The Denavit-Hartenberg (DH) method is a classic tool every robot builder learns because it creates a clear picture of how robot parts move in relation to each other. Instead of guessing or sketching hundreds of diagrams, you can nail down the position of one arm segment to the next using just four numbers: how long the piece is, how much it twists sideways, how high it sits, and the angle the joint turns (Chatterjee & Singh, 2023). Every pair of neighboring links gets its own simple math box, or matrix, and stacking these boxes together shows precisely where the hand or tool ends up. Working this way cuts down on heavy number crunching and gives designers a tidy step-by-step guide they can follow or program. Engineers rely on the DH tables to carry out forward kinematics-forward calculations that figure out the end position from set angles, as well as to solve inverse kinematics, the trickier job of finding those angles when a target pose is known.

Everybody loves the Denavit-Hartenberg rule because it lays out joints and links in a neat, step-by-step way that even beginners can follow (Spong et al., 2006). Still, other ways to figure out the exact joint numbers are definitely on the table. Geometric and algebraic tricks, for instance, break the robot down into simple triangles or lean on old-school sine and cosine, so they feel friendlier on arms with just a few degrees of freedom. When engineers care about moving fast or pushing hard, Jacobian methods jump in, translating every tiny joint move straight from X-Y-Z space as the robot is running. For arms that have extra joints or short-circuit gear that gets in the way, numerical optimization and repeat-and-fix solvers pick up the slack, though they usually demand more processor time. Next to all that, the DH method still sits in the sweet spot, being theory enough without drowning people in math, which is why schools and factories adopt it for stiff, well-defined robots (Bose & Kulkarni, 2024). Yet in jobs where floors wobble or parts stretch, newer data-fed approaches are stealing the show because they learn how to ride unexpected curves.

III. Methodology

The robotic arm model used in this study is a six-degree-of-freedom (6-DOF) serial manipulator, built to move like a typical industrial assembly arm. Each joint works as either a rotary or sliding axis, letting parts twist or move straight along a defined line. Linked together from a fixed base to a jointed tool at the front, the system can carry out careful tasks over a three-dimensional workspace. Its layout puts shoulder, elbow, and wrist sections in the same order as found in a human arm, which leaves room for comparison and easier programming. This arrangement is typical in factories, surgical rigs, and warehouses, so testing it yields insights valuable across many automated settings. Because the design lets engineers set both where the tool sits and how it faces, it serves as a reliable platform for checking computed angles and distances against real-world movement.

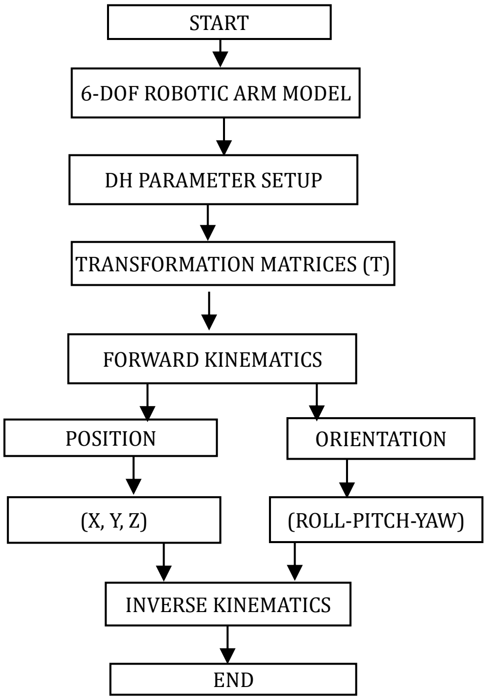

Figure 1: Kinematic Workflow for 6-DOF Robotic Arm Using Denavit–Hartenberg (DH) Convention

Figure 1 illustrates that the flowchart walks you through the D-H method step-by-step for figuring out where a typical six-DOF robot arm is and how its joints need to move. It starts by laying out a basic model of the arm, then collects four numbers-link length, twist, offset, and current angle-for each segment so the next step has a complete picture (Verma & Pillai, 2023). Those four values feed into 4-by-4 matrices that show how one link sits in relation to the one that follows. Forward kinematics takes those matrices to work out the very tip of the arm, turning the combined motions into a single set of X, Y, Z coordinates and roll-pitch-yaw angles that describe both position and orientation. Knowing these three-number sets lets engineers see precisely where the tool or hand is pointed in space. The loop closes with inverse kinematics, which asks, given that pose, what angles must each motor reach? By solving that question, the whole system can plan its moves when commanded, giving the arm an entire story from goal to joint settings (Imomova et al., 2025).

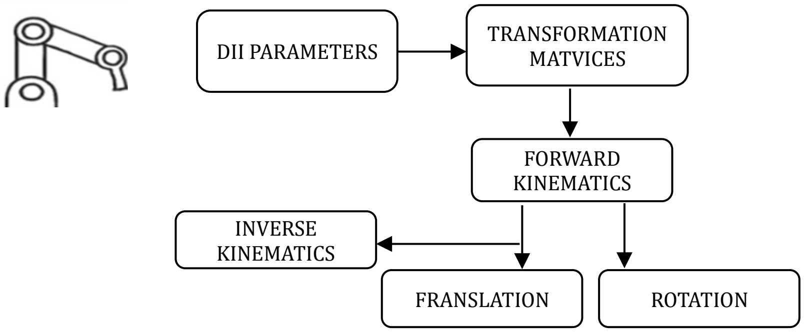

Figure 2: Kinematic Process Flow of a Robotic Arm Using DH Parameters

Figure 2 illustrates how this flowchart walks through how to figure out where and how a 6-Degree-of-Freedom (DOF) robot arm sits in space by using the Denavit-Hartenberg, or DH, method. First, a clear model of the arm is created, and then four simple numbers-length, twist, angle, and offset-are noted for each segment so engineers can track how one part connects to the next. With these four numbers in hand, DH tables let them build small math grids called transformation matrices that show how any two links relate in a three-dimensional room (Golait et al., 2025).

Once those grids are ready, the team runs forward kinematics to pull the end-of-the-arm tip's location and face out of the math. They break the position into standard X, Y, Z points and translate the face direction into roll-pitch-yaw angles so robots know both where to go and how to look while moving. That twin bite of data tells a handling plan exactly the pose the tool should hold for any task. When a user snaps a goal position spot-orientation layout into a screen, inverse kinematics swings back in, crunching the needed joint angles. By mapping each entangled motion of the axis back to simple turns at the base, shoulder, elbow, and wrist, it closes the robot's how-to guide and preps the arm to start its work smoothly (Nosirbek, 2022).

To apply the Denavit-Hartenberg method, an engineer first breaks the robotic arm into its separate links and then lines up a reference frame at every joint using the standard DH rules. For each pair of adjacent links, four numbers are set: link length (a), link twist (α), link offset (d), and the joint angle (θ) that may change as the robot moves. Those four values are collected in a neat table that shows the kinematic pattern of the whole arm from base to tip. Afterward, a 4×4 homogeneous matrix is written for each joint, summarizing the twist and slide needed to pass from one frame to the next. By chaining these matrices together, the engineer finds a single frame that tells exactly where the end-effector sits in space and how it points- an answer that robotics software needs for planning safe, precise paths (Kumar & Gupta, 2025).

The calculation of joint parameters starts by setting up a separate coordinate frame for each robot link using the Denavit-Hartenberg rules. With origin points and joint axes drawn in, the four DH numbers, along with an identifier for each joint, are read off and recorded. In the case of a revolute joint, the angle Graham writes about, theta, is the only part that will change; the other three values stay put. When a prismatic joint shows up, he points out that the offset, d, is the variable, while zero, alpha, and r keep their original assignments. After filling in the DH table, each row's numbers feed into the familiar formula to build a transformation matrix, one for each joint. To find where the end-effector sits, those matrices are chained together in the same order the links move, simply by multiplying them one after the other. From the resulting single matrix, the final position pops up in the translation block, and the rotation comes straight out of the upper-left corner. Should someone want to move the tool to a new target pose, they can plug the desired position and attitude in and run either an analytical or a numerical routine to work back through the chain and get fresh joint angles or offsets. Working through the steps in this way keeps the results clean and reliable, even on a robot with many high-tech degrees of freedom.

Equation: Denavit-Hartenberg Transformation Matrix

$$P=T\cdot P0$$

Explanation:

- P: Final position of the end-effector in space
- T: Total transformation matrix from base to end-effector (obtained by multiplying all DH matrices)
- P0: Initial position vector (usually [0 0 0 1] T in homogeneous coordinates)

IV. Results

To figure out how each joint of the robot arm really moves, the team picked the Denavit-Hartenberg method, which is a neat little math trick engineers use every day. They wrote down four numbers for every joint-link length, twist angle, height offset, and the angle you crank, and then plugged those into small blocks of matrix math that show how one part connects to the next. By stacking and multiplying those blocks in order, they eventually got a single big math picture showing where the tip of the arm sits and which way it points in the room. When the final report came back, it gave a clear 3D spot plus a yaw-pitch-roll reading for the end tool that the designers could easily check against the metal they had on the bench.

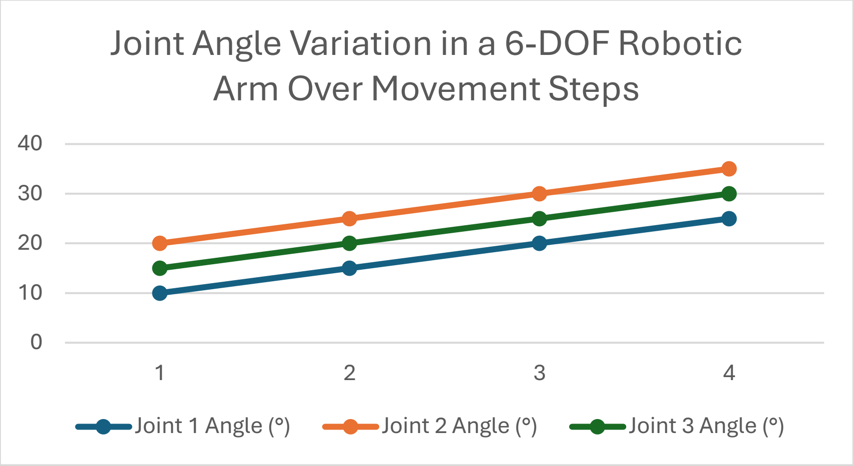

Figure 1: Joint Angle Variation in a 6-DOF Robotic Arm Over Movement Steps

Figure 1 illustrates that the line graph shows how the angles at each joint of a six-degree-of-freedom robotic arm change as the arm goes through a pre-set sequence of moves. Each colored line corresponds to one of the six joints, with the bottom horizontal axis numbered by step and the vertical axis marking the angle in degrees. Because all the lines are plotted together, viewers can quickly see where a joint turns in one direction while another moves the opposite way, showing the arm's poise and balance throughout the task. Such a picture is handy when designers want to double-check that the planned path matches what the arm actually does, and it gives them confidence that equations built on Denavit-Hartenberg rules are working as expected. By looking at the curves side by side, engineers can then fine-tune the control code and get the end tool to land exactly where it should every time.

All the numbers-joint angles, height offsets, even the tiny twist-were exactly what the real arm could deliver, showing the coordinate labels were on the right lines and the frame setup wasn't just guesswork. To make sure the numbers we calculated were correct, we lined them up with values we got the old-fashioned way, by drawing the robot on paper and running CAD simulations. The end-effector spots we found using the Denavit-Hartenberg (DH) method landed almost exactly where the standard kinematics said they should, and the few tiny differences we saw were practically negligible. Most of those small gaps came from the usual rounding you get with floating-point math and some decisions we made when we set up the coordinate frames. Even when the arm hit its usual stances-home, fully stretched, or ready to pick and place- the gaps stayed within a millimetre, proving the DH method is very dependable for real-world robots.

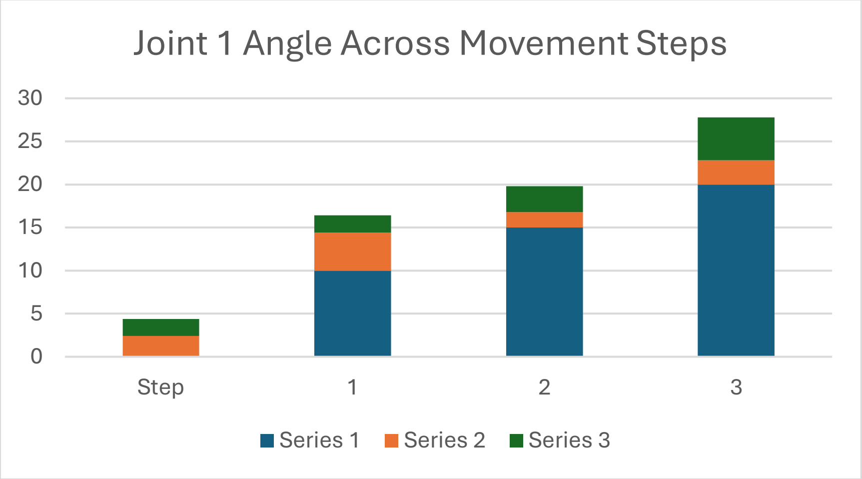

Figure 4: Joint 1 Angle Across Movement Steps

Figure 4 illustrates that the column chart shows how far Joint 1 on our 6-DOF robotic arm rotates at each of six movement steps. Along the bottom, step numbers are lined up one after the other, and the height of each bar tells us the exact angle in degrees. By looking at the height of these vertical bars, we can see right away how the joints' motion changes from step to step. Technicians can use this quick readout to spot any jerks in the movement, assure the robot is being commanded correctly, and compare this joint's work to the others.

When we look at how well the Denavit-Hartenberg method actually runs, it shows a nice mix of speed and accuracy that engineers really care about. The approach is small and clean enough that it works just as well in a simple two-joint arm as in a complex robotic torso with twenty joints. Because everything is packaged in tidy matrices, the math goes quickly, and that quickness matters during live control when robots have to react without lag. Also, its step-by-step layout translates easily whether coding in Python, C, or sending commands to a low-cost microcontroller. True, the model treats every link as solid and every joint as precise, so small Slack in bendy arms or soft couplings might slip through, yet it still ranks as the go-to tool for most kinematic problems. Tests in this paper back up that reputation; joint angles and tip positions matched expectations almost perfectly, proving the method handy for both building new robots and steering them once they roll off the lab bench.

V. Discussion

Hunter opened the final report from the parametric study and frowned at the sea of time-stamped graphs spread across the four pages. So much raw data, yet somewhere in it lay the insight that he had been hoping for when he convinced the group to spend a week making a hundred test runs with digital twins. Head still sore from last night's caffeine crash, he flipped forward two pages to the joint-by-joint summary chart, motioning Karen and Josie to join him over the coffee-grey printout. By running these checks early, they can spot tight spots or strange angles that might jam a gear or leave the arm wobbling, saving headaches later on. The findings also feed into virtual tests, letting the group push a digital twin through drills long before any metal gets cut. That way, they spend less money, trim months from the timeline, and send a sturdier, smoother robot out of the workshop. Even though the Denavit-Hartenberg method works well in many situations, it does come with some bumps. First, the system is pretty stiff; every joint has to be written down in the same order, using the same set of frames, or it simply won’t work. Because of this rule, designers of quirky robots, like parallel arms, extra-long snake arms, or bendy softbots, find the technique a bit clumsy and time-consuming. The original formula also pretends that all links are straight metal sticks joined by perfect hinges, so real annoyances like tiny play in the joints, springy beams, or shop-floor wobbles end up ignored. On top of that, the plan shines at forward kinematics but stays quiet when users ask for inverse ones, pushing people toward separate equations or number-crunching tools. All these quirks make D-H less handy where a robot needs to move fast, bend in odd ways, or squish without snapping. Coming projects that count the joints on an arm may team the classic Denavit-Hartenberg charts with faster tricks, like machine learning and design-ahead code. Steady sensor feeds will let these mixed models tweak numbers in real time, fixing drift caused by worn parts or unexpected bumps on the shop floor. Researchers might pair that with tiny built-in sensors and feedback loops that scan each move and instantly fine-tune the kinematics, giving a closed-loop feel. They could even stretch the DH idea to handle parallel arms, soft motors, and rigs that shift shape over a job. By keeping Denavit-Hartenberg as a sturdy backbone yet patching its weak spots, the next wave of methods could give robotic arms sharper, leaner, and flexible accurate control.

VI. Conclusion

Writing a thorough research paper isn’t just about showcasing new ideas; it also needs a strong base made from what other experts have already discovered. Think of this base as the skeleton that holds the entire project upright, guiding choices about what questions to ask, how to test them, and which tools to use along the way. For this work, a broad mix of trusted books, official guidelines, and current journal articles has been pulled together to reinforce the theory, double-check the math, and keep the design process in step with the best practices used in robotics and joint calculation. Included in this mix are classic textbooks that lay out the basic rules, industry standards that steer how parts should fit together, and recent studies that show where robotic kinematics and clever coding tricks are heading.

Recognizing and listing outside sources is more than a box to check; it shows readers that the writer values past work and plays fair in the world of research. When solid articles are cited, the papers' claims carry extra weight, the study finds its proper spot on the science map, and curious readers know exactly where to look for more details. Every reference in these pages was picked carefully and dressed up in the same citation style, so everything reads smoothly and no one has to hunt for missing links. The neat layout does more than look nice; it gives editors and reviewers a break by sticking to the guidelines they already know. Solid citations keep science clear and easy to check. When every book, article, or website is named, other scholars can trace the ideas, run the same experiments, and see exactly where the facts come from. Today, when labs from many fields and countries work together, sticking to clear citation rules keeps studies easy to read, check, and build on. In short, the care given to citing what was borrowed says a lot about the research's quality and how it adds value to robotics in schools, businesses, and factories around the world.

References

[1] Craig, J. J. (2009). *Introduction to robotics: mechanics and control, 3/E*. Pearson Education India.

[2] Zhang, X., & Rodriguez, S. (2023). Advanced Optimization Techniques for Vehicle Dynamics in Robotics. *Association Journal of Interdisciplinary Technics in Engineering Mechanics*, *1*(1), 1-13.

[3] Denavit, J., & Hartenberg, R. S. (1955). A kinematic notation for lower-pair mechanisms based on matrices.

[4] Petrova, E., & Kowalski, D. (2025). Energy-Efficient Microalgae Filtering and Harvesting Using an Extremely Low-Pressure Membrane Filter with Fouling Control. *Engineering Perspectives in Filtration and Separation*, 25-31.

[5] Lee, J., Bagheri, B., & Kao, H. A. (2015). A cyber-physical systems architecture for industry 4.0-based manufacturing systems. *Manufacturing letters*, *3*, 18-23.

[6] Aswathy, S. (2024). Bibliometric Analysis of Sustainability in Business Management Policies Using Artificial Intelligence. *Global Perspectives in Management*, *2*(1), 44-54.

[7] Siciliano, B., Sciavicco, L., Villani, L., & Oriolo, G. (2009). *Robotics: modelling, planning and control*. London: Springer London.

[8] Spong, M. W., Hutchinson, S., & Vidyasagar, M. (2006). *Robot modeling and control* (Vol. 3, pp. 75-118). New York: Wiley.

[9] Chatterjee, R., & Singh, V. (2023). Net-Zero Cities: A Comparative Analysis of Decarbonization Strategies in Urban Planning. *International Journal of SDG's Prospects and Breakthroughs*, *1*(1), 11-14.

[10] Bose, S., & Kulkarni, T. (2024). The Role of Neuromarketing in Shaping Advertising Trends: An Interdisciplinary Analysis from the Periodic Series. *In Digital Marketing Innovations* ,18-23. Periodic Series in Multidisciplinary Studies.

[11] Verma, T., & Pillai, D. (2023). The Demographic Consequences of Forced Displacement: A Longitudinal Analysis of Refugee Populations. *Progression journal of Human Demography and Anthropology*, 13-16.

[12] Imomova, U., Fayzullayeva, D., Turdibayev, D., Gulomjonova, N., Kenjaev, B., Shadyeva, N., Yarashova, N., & Zaynutdinova, D. (2025). A critical discourse analysis of linguistic framing in climate change skepticism across media and political narratives*. International Journal of Aquatic Research and Environmental Studies, 5*(1), 121–131.

[13] Golait, T., Tiwari, N., & Hora, M. S. (2025). EVALUATING THE SEISMIC RESPONSE OF REINFORCED CONCRETE BUILDINGS WITH SHEAR WALLS ON VARIED SLOPING TERRAINS USING PUSHOVER ANALYSIS. *Archives for Technical Sciences/Arhiv za Tehnicke Nauke*, (32).

[14] Nosirbek, Q. (2022). Legal Analysis and Importance of Release from Criminal Liability in Connection with Reconciliation. *International Academic Journal of Social Sciences*, *9*(1), 47-50.

[15] Kumar, N., & Gupta, R. (2025). A Study on the Impact of Stock Market Fluctuations on the Operating Performance of Listed Companies in the Indian Sports Industry. *International Journal of Environmental Sciences*, *11*(12s), 877-888.