This paper is a preprint (IEEE “accepted” status).

DOI. 10.1109/ISETC63109.2024.10797341

Simulating Human Physical Behavior using Unreal Engine Full Body Inverse Kinematics

Alexandru-Mihai Pescaru
Department of Computer and Information Technology
Politehnica University Timisoara
V. Parvan 2, 300223 Timisoara, Romania
alexandru.pescaru@cs.upt.ro

Razvan Adrian Bertici
Department XIII Pulmonology
"Victor Babeș" University of Medicine and Pharmacy Timișoara
Eftimie Murgu Sq. No. 2, 300041 Timișoara, Romania
razvan.bertici@umft.ro

Traian-Radu Plosca
Department of Automation and Applied Informatics
Politehnica University Timisoara
V. Parvan 2, 300223 Timisoara, Romania
traian.plosca@aut.upt.ro

Mihai Victor Micea
Department of Computer and Information Technology
Politehnica University Timisoara
V. Parvan 2, 300223 Timisoara, Romania
mihai.micea@cs.upt.ro

***Abstract*—Extracting, analyzing, and simulating human physical behavior have proven to be challenging tasks for researchers all over the world for many years. If developed successfully, they can generate important results that will contribute to progress in many fields. These results can change many people's lives, aiding in a handful of domains, such as security, medicine, autonomous robots, and many more. A direct outcome is to obtain the data necessary to train and validate advanced neural network models implied by many of today's solutions. We propose here an approach that uses Unreal Engine 5 to reproduce the movements characteristic to normal human behavior in a simulated environment, considering restrictions of body joints. This method can be used to aid in creating flawless body animation, both procedural and AI generated. It can be included in healthcare-related body simulations, video game development, and in some other fields involving research on physical human or animal behavior.**

***Index Terms*—visual simulation, behavior analysis, range of motion, inverse kinematics, robotics, video games.**

I. Introduction

Powered by recent advances in hardware and machine learning technology, the development of revolutionary tools that can be used in security, automotive, medicine, and other domains has become a priority for researchers all over the world. Great effort was focused on the analysis and simulation of human physical behavior, which has become a hot topic in the last decade. The reproduction of human behavior can be a difficult task, due to varied responses when different subjects are faced with similar scenarios. This problem has basically two solutions, either to consider a common observed response or to train a model based on analyzing a huge amount of data. Most of the algorithms used in behavior simulation imply at some point a machine learning model, either to extract or to analyze behavior, or for both. In all scenarios, training and validating such complex models involve a lot of data. Therefore, one of the most challenging issues in this case is how to obtain the necessary data in a short time and at a reasonable cost.

Moreover, considering such advanced artificial intelligence models trained, the problem of testing and validation of applications still remains. In many cases, it implies expensive objects and machinery. Especially in the early stages of research, testing algorithms on robots, or even dummies, can be costly due to higher rates of errors and failures. Therefore, testing an algorithm or the continuous training of an artificial neural network can be done cheaper when using a simulation environment that can mimic the real-life behavior of entities involved in the scenario. Developing such simulators is also a challenging task, especially when we consider algorithms that process images or video sequences. A commonly adopted solution in this case is to implement a simulator based on game engine software that can produce graphics that are faithful to reality, such as Unreal Engine [1] or Unity [2].

Our research is focused on the possibility of integrating Unreal Engine version 5 into a human physical behavior simulator. It is a game engine developed by Epic Games known for its variety of features, photorealistic graphics capabilities, and advanced optimizations. We choose this version determined by the integration of Full Body Inverse Kinematics (FBIK). FBIK was implemented in Unreal Engine 5 based on the PowerIK [3] plugin available in the previous version and offers the possibility to apply inverse kinematics on a chain of bones, without a limit for the chain length, or even on multiple chains with a common linking point.

II. Related Work

There are multiple approaches used to simulate human natural movements, but they differ hardly in results. They spread from manual character animation based on the reproduction of an observed behavior of a real entity to *retargeting animations* aiming to simulate the movement by reusing it between characters. The common approach is to produce a character in a virtual world that moves quasi-naturally and use it for film production, video games, robotics, medical applications, and in general in domains that require behavior analysis and simulation.

Creating manual animations is not an easy task. An artist needs years of experience to achieve realistic and good looking results. Important research was spent in developing techniques to help artists. Zhang et al. [4] propose an automatic method to analyze a 3D animation designed to assist the creator. The methodology starts by analyzing the artist's work, and then proposes corrections and improvements. The algorithm is targeting especially students who are learning to animate in 3D, helping them to achieve faster and better results. Mourot [5] identifies the same difficulties in manual 3D animation. He proposes an algorithm which can analyze the 3D animation scene and adjust joint transforms when needed. The tools provided consider the position and orientation of joints and, based on real-world information extracted, try to correct different aspects of the animation. The algorithm solved issues such as incorrect joint transforms or incorrect foot placement.

Creating the animation from scratch requires time and consumes important resources. Therefore, generating or reusing animations proved to be helpful approaches to faster and cheaper development. Many investigations have been conducted in this direction over the years [6]. Most recent methods use machine learning as an important development tool that proves to solve many problems that are too complex for a procedural approach. Lin and Lee [7] propose a method to analyze the pose in 2D videos and generate 3D animations using deep learning. The animation is generated on the basis of a sequence of trajectories. Another method is proposed in [8] for animation retargeting. They use a vision language model that analyzes both the skeleton and the mesh of the source to complete the task.

In addition to their original purpose, game engines are widely used in simulation behavior research. A well-known example is the Microsoft AirSim developed by Microsoft that is used to easily implement simulators based on its flying physics support. Many researchers also used Unity and Unreal Engine [9] for a variety of reasons, such as their out-of-the-box rendering capabilities, advanced physics simulation, and easy-to-use architectures.

Starke et al. [10] integrated Unity3D into a robot simulator. They developed a FBIK system to simulate robotic arm movement and to easily control robots, either by AI or by using hardware gadgets. Lam et al. [11] are researching the possibility of developing a large-scale metaverse application, a place where people express themselves in a virtual space. They developed a system that could extract and reproduce human movement for avatars using Axis Studio for motion capture, and Unity3D to model the metaverse.

III. Articulated Body Motion

An important challenge in the simulation of articulated body motion is represented by the large number of degrees of freedom considering all segments of the limbs. To cope with this problem, most solutions use a model of the body built around a framework of a kinematic chain, as seen in Fig. 1.a.

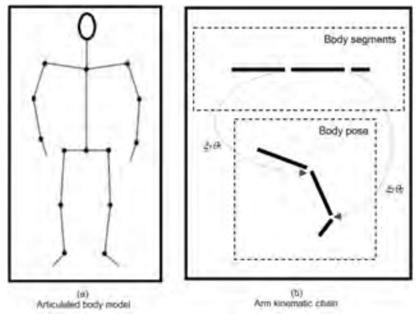

Fig. 1. Articulated body as a kinetic chain.

The chain consists of 15 to 17 segments, depending on the inclusion of the wrists. Up to six degrees of freedom result from translation and rotation, and three degrees of freedom from shoulder and hip joints. The clavicle joints are given two degrees of freedom since we do not consider here rotation on their own axis, and an extra degree from the remaining joints considered as hinges. Therefore, we can consider up to 29 degrees of freedom.

The problem of describing the articulated body motion can be represented using a transformation of a linear body frame to map a specific body pose and a motion vector representing the velocities of body segments. As an example, we take an arm represented by three segments and two joins as depicted in Fig. 1.b. Each joint defines an axis of rotation described by a 3D unit vector ω_1 along the axis and a point q_1 on the axis.

This hinge joint can be modeled by a twist ξ_1 (1). A rotation of angle θ_1 around the joint axis can be expressed as $e^{\hat{\xi}_i \theta_i}$ [12].

$$\xi_1 = \begin{bmatrix} -\omega_1 \times q_1 \\ \omega_1 \end{bmatrix} \tag{1}$$

For a set of k segments, the k^{th} joint position is described by a twist ξ_k and a rotation angle θ_k. The velocity of each segment k in the kinematic chain is described by a twist vector V_k, which represents a linear combination of segments twists and angular velocities (2).

$$V_k = \xi_1{}'\dot{\theta_1} + \xi_2{}'\dot{\theta_2} + \cdots + \xi_k{}'\dot{\theta_k} \tag{2}$$

For an arbitrary point q_c in the k^{th} segment, the motion vector can be expressed as (3).

$$\begin{bmatrix} u_x \\ u_y \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \end{bmatrix} \begin{bmatrix} \hat{\xi_1}{}'\dot{\theta_1} + \hat{\xi_2}{}'\dot{\theta_2} + \cdots + \hat{\xi_k}{}'\dot{\theta_k} \end{bmatrix} \tag{3}$$

Based on that geometric calculation, heuristic methods for inverse kinematics, such as the one implemented by Unreal Engine, will estimate the joint updates to produce the animation. The technique is known as *forward and backward reaching inverse kinematics* algorithm and shows good performance with relatively low computational requirements [13]. It consists of two consecutive phases in an iterative forward-backward manner. An important advantage is the global optimization applied to all joints in one step.

IV. THE PROPOSED APPROACH

In this paper, we propose a methodology for integrating Unreal Engine 5 in a human physical behavior simulator. A rigged mesh animated using Unreal Engine 5 Control Rig objects and FBIK is employed in motion reproduction. In addition, a low poly mesh was created and rigged for testing and enhanced visual feedback. The skeleton used in the experiments was modeled after the standard proposed by Unreal in their tutorial. We adopt the original layout and naming convention but adapt it to our mesh dimensions. Fig. 2 presents the model, imported into Unreal Engine 5.

Control Rig objects support multiple nodes of different types, such as IK nodes and FBIK nodes. Therefore, multiple layouts were developed and tested to see how the model behaves. The reference node used in our work was FBIK. Experiments using such nodes were conducted to find the most realistic generated movement. Few additional experiments were performed with other types of nodes for comparison, involved movements of the head and neck. We learned that using a single FBIK instance that controls the entire skeleton can cause important problems, such as random and unnatural movements of the bones. Better results were obtained when we used multiple FBIK instances. We conclude that they can create better reproduction of natural movement in case of using the current implementation of FBIK in Unreal Engine 5.

The resulting layout developed to test our approach is made up of four control groups described below:

- A group for the legs. This group contains two FBIK chains that start in the pelvis and end at the toe bone for both legs. Each chain has the hip bone as the root for the FBIK chains, which model the leg and the tight bones.
- A group for each arm. Each FBIK controls a chain of four bones, having the root as the right clavicle, the respective as the left clavicle, and ending with the metacarpals for each arm.
- A group for the spine. It is controlled using a *Distribute Rotation* node. This node takes a rotation and distributes it among bones in the chain. The chain is composed of the spine, together with the neck and head bones. The default function used to implement rotation is linear.

To simulate natural movements, free of mesh clipping, jarring movements, or issues derived from the shaking of mesh parts, we reviewed medical studies [14]–[16] on the range of motion of aligned joints. The primary concern was the upper extremities due to their magnitude and variety of motion, which include flexion, extension, supination, pronation, abduction, and adduction. Based on these studies, the improved model movements were smooth and free from significant clipping problems and other aforementioned issues of previous iterations.

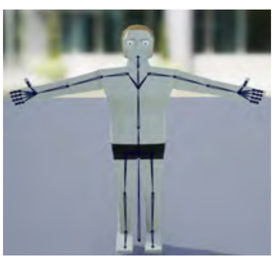

Fig. 2. The skeleton model with the placeholder mesh.

V. EXPERIMENTAL ANALYSIS

For the experimental analysis, we develop a prototype following the presented approach implementing a small character based video game. It includes a Control Rig with FBIK used to generate procedural animations of humanoid characters. The physical behavior of the characters was implemented as a set of scripts inspired by Unreal documentation, where a dinosaur model is proposed as an example of FBIK integration. We adapted this solution for humanoid characters by making the necessary adjustments to simulate persons walk and run activities, and by adding the possibility to vary the walking speed. The implementation considers the relationship between the speed of the person and the dynamics and the horizontal displacement of the feet. Additional control is used at the pelvis level to simulate crouching, jumping, and other movements that involve the whole body. The FBIK support control over the coordinates of all other control points.

To simulate the movement of hands, we adopted a solution based on directional targeting inspired by games like Mordhau and Bannerlord. This type of mechanics is specific to combat actions controlled using a mouse or joystick. The hand moves in a plane on the chosen trajectory, which is described relative to the origin of the character and the direction the camera is looking.

Our prototype uses ten control points per character. Of the first eight points, two correspond to horizontal trajectories, four to diagonal trajectories, and two to vertical movements. Additionally, a stab motion requires only two additional points since it always occurs on the same trajectory on the right side of the body. Each hand can traverse the trajectories in both directions by changing the orientation, except for the vertical and stabbing movements, where it makes sense only in one direction. The localization of control points is shown in 3, where the red marks represent the extremities of the trajectories, the blue marks indicate the margin on the right side, the green delimits the left side and the yellow is used for pelvic control. The orientation of the hand is determined by the starting and ending positions, and is interpolated on the trajectory between the two points, as described in Algorithm1.

For subjective evaluation, we used three virtual cameras to observe the experiments. A first-person camera, which captures what the character sees, a camera positioned in front of the character and a camera that captures the simulation from behind. This allowed us to conduct a comprehensive visual analysis of the trajectories of the hands and to appreciate the smoothness of the trajectories. Two categories of hand movement were evaluated, a hit and a reposition. The hit

Algorithm 1 Hand movement simulation

$\vec{p_1} \leftarrow$ Translation of the starting point
$r_1 \leftarrow$ Rotation of the starting point
$\vec{p_2} \leftarrow$ Translation of the finish point
$r_2 \leftarrow$ Rotation of the finish point
$et \leftarrow$ Elapsed time
$\vec{q_1} \leftarrow Quaternion(\hat{p}_1 \times \hat{p}_2,\ 0)$
$\vec{q_2} \leftarrow Quaternion(\hat{p}_1 \times \hat{p}_2,\ acos(\hat{p}_1 \cdot \hat{p}_2))$
$\vec{q} \leftarrow Interpolate(\vec{q_1},\ \vec{q_2}, et)$
$i \leftarrow Interpolate(|\vec{p_1}|,\ |\vec{p_2}|, et)$
$\vec{vec} \leftarrow Scale(\vec{p_1},\ i \div |\vec{p_1}|)$
$\vec{translation} \leftarrow RotateVector(\vec{vec}, \vec{q})$
$rotation \leftarrow interpolate(r_1,\ r_2,\ et)$
* Apply the $\vec{translation}$ and the $rotation$ to the hand

represents any movements on any trajectory, vertical, horizontal, diagonal, and trust. Reposition refers to a translation of the hand from the current position to the starting position of the next action. This occurs when the next movement develops in another plane or direction than where the hand is at the moment. We employ three evaluators and the unanimous conclusion was "perfectly natural" and "no motion artifacts observed". Finally, we conduct a performance test on an AMD Ryzen 3700x PC with 32 RAM and a Radeon RX7900-GRE involving simulation of scenes consisting of 1 to 180 characters. The results are presented in Fig. 4 and demonstrate a quasi-linear rendering performance with the number of characters.

VI. Conclusions and Future Work

In this study, we proposed an approach based on Unreal Engine 5 to simulate human physical behavior. Our methodology uses the FBIK in the Control Rig objects. Multiple FBIK nodes control different parts of the body for natural movement simulation. A prototype was developed based on an animated character. It was used to analyze the performance of the simulator and assess the naturalness of limb movements in a subjective visual evaluation. In future development, we plan to integrate this prototype with a medical application to study the effect of bone injuries on joint mobility.

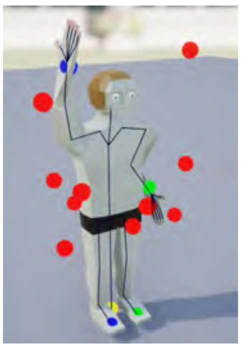

Fig. 3. The mesh and layout for the experiment.

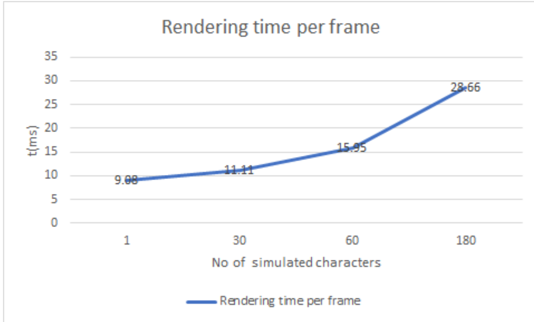

Fig. 4. Simulation rendering performance from crowd scenes (time/frame)

References

[1] Epic Games, "Unreal engine 5." [Online]. Available: https://www.unrealengine.com

[2] J. K. Haas, "A history of the unity game engine," *Worcester Polytechnic Institute*, 2014.

[3] Power Animated, Epic Games, "Power ik." [Online]. Available: https://poweranimated.github.io

[4] Y. Zhang, "Adjustment design of motion rhythm of 3d animation based on feature extraction and recognition," in *2020 IEEE Conference on Telecommunications, Optics and Computer Science*, 2020, pp. 121–124.

[5] L. Mourot, "Deep learning for skeletal character animation : topology editing, retargeting and cleaning," Theses, Université de Rennes, May 2023. [Online]. Available: https://theses.hal.science/tel-04219725

[6] L. Mourot, L. Hoyet, F. L. Clerc, F. Schnitzler, and P. Hellier, "A survey on deep learning for skeleton-based human animation," *CoRR*, vol. abs/2110.06901, 2021. [Online]. Available: https://arxiv.org/abs/2110.06901

[7] J. Lin and G. H. Lee, "Trajectory space factorization for deep video-based 3d human pose estimation," 2019. [Online]. Available: https://arxiv.org/abs/1908.08289

[8] H. Zhang, Z. Chen, H. Xu, L. Hao, X. Wu, S. Xu, Z. Zhang, Y. Wang, and R. Xiong, "Semantics-aware motion retargeting with vision-language models," 2024. [Online]. Available: https://arxiv.org/abs/2312.01964

[9] A. Jungherr and D. B. Schlarb, "The extended reach of game engine companies: How companies like epic games and unity technologies provide platforms for extended reality applications and the metaverse," *Social Media + Society*, vol. 8, no. 2, p. 20563051221107641, 2022. [Online]. Available: https://doi.org/10.1177/20563051221107641

[10] S. Starke, N. Hendrich, and J. Zhang, "Memetic evolution for generic full-body inverse kinematics in robotics and animation," *IEEE Transactions on Evolutionary Computation*, vol. 23, no. 3, pp. 406–420, 2019.

[11] K. Y. Lam, L. Yang, A. Alhilal, L.-H. Lee, G. Tyson, and P. Hui, "Human-avatar interaction in metaverse: Framework for full-body interaction," in *Proceedings of the 4th ACM International Conference on Multimedia in Asia*, ser. MMAsia '22. New York, NY, USA: Association for Computing Machinery, 2022. [Online]. Available: https://doi.org/10.1145/3551626.3564936

[12] M. R.M., L. Z., and S. S.S., "A mathematical introduction to robotic manipulation," *CRC Press*, 1994.

[13] A. A., C. Y., and L. J., "Extending fabrik with model constraints," *Computer Animation and Virtual Worlds*, vol. 27, no. 1, pp. 35–57, 2016.

[14] M. F. Rybski, "Range of motion," in *Kinesiology for Occupational Therapy*. Routledge, 2024, pp. 37–65.

[15] D. H. Gates, L. S. Walters, J. Cowley, J. M. Wilken, and L. Resnik, "Range of Motion Requirements for Upper-Limb Activities of Daily Living," *The American Journal of Occupational Therapy*, vol. 70, no. 1, pp. 7001350010p1–7001350010p10, 12 2015. [Online]. Available: https://doi.org/10.5014/ajot.2016.015487

[16] J. Aizawa, T. Masuda, T. Koyama, K. Nakamaru, K. Isozaki, A. Okawa, and S. Morita, "Three-dimensional motion of the upper extremity joints during various activities of daily living," *Journal of Biomechanics*, vol. 43, no. 15, pp. 2915–2922, 2010. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0021929010003878