IEEE Access
Multidisciplinary ⋮ Rapid Review ⋮ Open Access Journal

Received 10 May 2024, accepted 12 June 2024, date of publication 20 June 2024, date of current version 5 July 2024.

Digital Object Identifier 10.1109/ACCESS.2024.3417220

RESEARCH ARTICLE

Lightweight Physics-Based Character for Generating Sensible Postures in Dynamic Environments

BIN HU[1], **HUI GUO**[1,2], **YULING YANG**[1], **XIONGJIE TAO**[1], **AND JIE HE**[3]

[1]Faculty of Arts and Humanities, Macau University of Science and Technology, Macau, China
[2]Guangxi Key Laboratory of Machine Vision and Intelligent Control, Wuzhou University, Wuzhou 543002, China
[3]College of Computer Science and Electronic Engineering, Hunan University, Changsha, Hunan 410082, China

Corresponding author: Hui Guo (3220002921@student.must.edu.mo)

This work was supported by the National Natural Science Foundation of China under Grant 62162054 and Grant 61961036.

ABSTRACT Pose change and environmental interaction have been the primary directions of recent research in the field of physics-based characters. To address the problems of a high production threshold, high real-time arithmetic consumption when combined with deep reinforcement learning, and a lack of realism in existing solutions, in this paper, a real-time automatic motion control model is designed for virtual characters that provides posture control through physical response linkages and that can generate smooth and natural interactions with the environment through small amounts of body adjustments. The proposed model consists of a mixture of kinematic controllers and lightweight physical interaction modules. It uses a designed posture correction module to correct the posture of a hybrid model, automatically adjusting the body posture, and it can respond reasonably to the environment when the action generation encounters environmental obstacles. The designed constraint scheme can reduce the tuning of joint parameters while allowing a character to have realistic human-like motion perception. The proposed model is verified by experiments, and the experimental results demonstrate that the proposed virtual human model can be adjusted to achieve the human-like effect using real action data and keyframe animation. Ball bouncing and walking on irregular terrain experiments verify that the proposed virtual character model can interact with the environment effectively in real-time. As the carrier of character animation, the proposed driving framework model can rapidly and easily generate interactive motions based on given input parameters.

INDEX TERMS Physics-based character, inverse kinematics, hybrid modeling, joint restraint.

I. INTRODUCTION

Virtual characters have been widely used in various fields, including digital entertainment, virtual manufacturing, and complex skills training. With the development of geometric morphological modeling and rendering drawing technologies, the requirements for realism, interactivity, and real-time performance of a virtual character's motion pose changes have increased significantly [1], [2], [3]. The motion generation and control technology of virtual characters represents the basis for realizing the pose changes of virtual characters and is directly related to the final visualization of virtual characters' behavior expression and cognitive expression [4]. Since the real-time generation of virtual character actions requires a large number of joint transformation computations, as well as meeting the requirements of various virtual scenarios, specific controlling inputs, and business constraints, a limb-driven architecture and a linkage mechanism of a virtual character have been of particular importance.

Currently, virtual characters commonly adopt a limb-driven architecture that combines skeletal systems and inverse kinematics (IK). The motion matching method, which has been widely employed in the electronic gaming industry, is commonly used for generating virtual character

The associate editor coordinating the review of this manuscript and approving it for publication was Songwen Pei.

motions [5]. Subsequent research has shown that the fidelity of motion matching is directly proportional to the size of the motion database, whereas memory usage and matching speed are inversely proportional to it [6].Motion matching necessitates more computational resources and time for processing intricate actions and scenes. Moreover, the performance of the system heavily relies on the quality and diversity of training data. Therefore, for applications demanding high precision and realism, the system requires further refinement and adjustment. Furthermore, the system's environmental awareness and decision-making abilities might be inadequate in complex environments or during character interactions. Nevertheless, in domains such as sports training and military simulations involving human-machine efficiency analysis, virtual characters often undertake specific and intricate physical simulation tasks. Consequently, their motion generation must combine strong environmental adaptability, visual realism, and physical fidelity.

The application of physics-based characters can address the problem of environmental responsiveness, and currently, training them in combination with deep learning and reinforcement learning can achieve good human-like motion effects [7], [8], [9], [10]. Despite the potential of physics-based characters, their widespread adoption has been hindered by several factors. These include the stability issues of the PD controller, the high computational costs involved [11], and the complexity inherent in designing simulation solutions [12]. These challenges reduce the efficiency of training actions, limit task diversity, and necessitate significant computational resources and time. As a result, physics-based characters have yet to achieve broad utilization.

This work proposes a real-time motion control model of a virtual character (biped), which uses a physically responsive linkage pose control mechanism to realize smooth and natural interactions with an environment through a small number of limb adjustments. The proposed model consists of an IK controller and a lightweight physical interaction module and incorporates a posture correction module based on the FABRIK-ER algorithm. When encountering environmental obstacles during motion generation, a character can automatically adjust its body posture to respond appropriately to the environmental conditions, resulting in more realistic human-like motion effects.

To achieve the aforementioned functions, this study uses a real-time hybrid limb-driven architecture that integrates a rag-doll system with an IK control strategy for motion control and change articulation in both stressed and unstressed states. The limb linkage of a virtual character considers physical weights to ensure that the rendering of its posture is natural in an arbitrary environment. Due to the relatively higher control authority of the inverse kinematics mechanism over the joints of the skeletal system, interference can be generated during physical interactions. Therefore, the proposed design automatically corrects conflicts arising from rigid body detection interactions and IK position control instructions, eliminating limb jitter and self-penetration. Furthermore, loose constraints are introduced on the pole axis, ensuring that dynamic orientations of the mid-joints of the limbs align with natural laws during changes in a character's posture [13]. The proposed architecture, even without the introduction of artificial intelligence methods, can generate realistic motions for virtual characters. It combines the physical interactivity and precision of kinematic-driven movements while achieving a higher level of human-like body posture. Consider the example presented in Fig. 1, in which a virtual character walks in a low-dimensional environment; the blue and purple character model in the scene denote virtual character rendered from real human motion data, and the white and orange character model are virtual character rendered using the proposed model. As shown in Fig. 1, the proposed model can adjust the motion posture according to changes in the environment without causing model penetration.

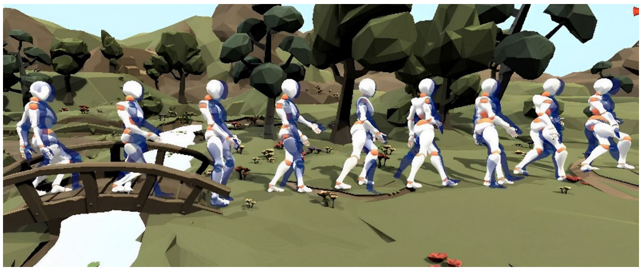


FIGURE 1. The virtual character walks in a low-dimensional environment. The blue and purple character models in the scene denote virtual characters rendered from real human motion data, and the white and orange character models in the scene represent virtual hybrid-architecture characters modified by the proposed solution.

The main contributions of this study are threefold and can be summarized as follows:

(1) A real-time hybrid limb-driven architecture that combines kinematic control strategies with physical feedback methods is proposed to control a character's action pose generation using the IK and secondary pose adjustment through environment detection by lightweight physical components. This enables a virtual character to generate smoothly different pose effects and interact physically with an environment under fewer input parameters and lower computational cost;

(2) An environment-responsive posture generation module, whose core algorithm is the FABRIK-ER algorithm, which is based on the FABRIK principle, is proposed. By adjusting the priorities, the action control is given lower priority compared to the environment response, thus avoiding the command conflict caused by the synchronized operation of the two control mechanisms and eliminating the phenomena of local model penetration and overall jitter of a virtual character during a hybrid drive;

(3) An elastic constraint method for elbow and knee orientation, which uses a spring-damping system to construct an elastic association between the pole and

the middle joint of the limbs, is introduced. The proposed method ensures dynamic rationality of the joint movement orientation with its natural orientation, making the movements more realistic and posture changes more in line with the characteristics of real human motion.

The rest of the paper is organized as follows. In Section II, previous work on virtual character action control methods is reviewed. In Section III, the scheme and architecture of the hybrid drive model proposed in this paper are explained.,and the attitude generation optimization scheme used in this model is discussed.and the principle of the proposed elastic constraint algorithm is explained. In Section IV, the motion control scheme of the proposed model is verified experimentally. Finally, in Section V, the advantages and limitations of the proposed scheme are discussed, as well as future work directions.

II. RELATED WORK

A. PHYSICAL-BASED CHARACTER CONTROL

With the continuous development of computer animation technology in recent years, three-dimensional (3D) game animation and movie special effect production have shown a highly increasing trend,character animation synthesis, as one of the main research topics of the mentioned industries, has attracted extensive research attention. Numerous methods have been proposed to implement character animation, such as using trigonometric functions to generate character animations [14], constructing upper-body animations based on lower-body animations [15], and motion matching techniques [16]. Among these methods, physics-based character control can synthesize actions and effectively interact with a virtual environment; therefore, it has been the primary focus of pose research in recent years.

The research on physics-based character control can be roughly divided into two parts: physics simulation and control. The physics simulation is not specific to physics-based characters. For instance, one of the deformable solid simulations is to simulate the rigid body components of a rag doll [17]. Like the unified Newtonian potential approach to multi-body dynamics [2], the physical simulation of virtual characters is simply one type of rigid body-related research. In [18], a system for (bipedal) humanoid models, which can rapidly generate physics-based characters, was proposed. Early studies on the control of physics-based characters were mostly aimed at robot behavior generation and modeling [19]. However, after the year 2000, computer animations became increasingly prominent. In the initial stages, researchers focused mainly on studying the basic behaviors of virtual characters, particularly on maintaining balance in various situations and the corresponding motion feedback [20], [21], [22], [23], [24], [25], [26], [27]. In [20], a composite controller framework was proposed, while in [28], motion styles were modified by adjusting muscle parameters. In [29], various motion behaviors were generated using optimization constraints. In [30], the authors designed specific generalized controllers for falling and landing scenarios. However, these controllers require redesigning and fine-tuning for different motions and models, which pose significant limitations to their practical applications. These controller schemes are not universal and require different designs and adjustments for different tasks, posing significant limitations. Therefore, when the Stable Proportional Derivative (SPD) controller was designed [31], the PD controller emerged as the most versatile controller for physics-based characters and was trained using artificial intelligence methods. References [32], [33], [34], [35], [36], and [37] all adopt the PD controller as the cornerstone of action control. However, the PD controller operates within a force-based framework, and the force applied is not solely derived from the PD controller's input but is also influenced by external forces from the environment. Consequently, the PD controller has always faced stability issues. Compared to methods for controlling non-physical virtual humans, it requires consideration of not just the action strategy, but also the rationality of the force application process, which demands more computational power. Therefore, its application in complex control strategies requires careful deliberation. Even though certain improvements in the PD controller's structure [37] can provide partial optimization, they cannot completely resolve the instability problem. Therefore, this study does not use a PD controller and instead employs an inverse kinematics embedding approach as a behavioral controller for generating motion.

B. HYBRID CHARACTER MODEL USING KINEMATICS

Kinematic-driven virtual characters can efficiently compute postures and perform coordinated adjustments for limb movements. There is a wide variety of IK algorithms [38], which are extensively used in the motion control of robots. Some studies [39], [40] even combined multiple IK solutions to form a general approach. In recent years, the IK algorithms have been continuously optimized [41], [42], [43] to achieve the best performances in terms of computational efficiency and optimal solutions. Currently, the application of kinematic-blend character models has shown good results in various motion postures. For instance, the authors of [44] combined neural networks to generate lifelike motions. In [45], the authors combined collision handling with spatial relationships in the environment. The authors of [46] used neural network algorithms to generate interactive animations, and in [3], a combination of kinematics and dynamics was used to solve upper body posture and interaction problems.

References [3] and [46] both adopt a method that combines kinematics and a physical torso. It is evident that if the interaction problem between kinematics and the physical torso can be resolved, it would result in impressive interactive effects. Therefore, this study uses the IK as a behavior controller of physical characters to realize action linkage control and respond to environmental changes. Referring to

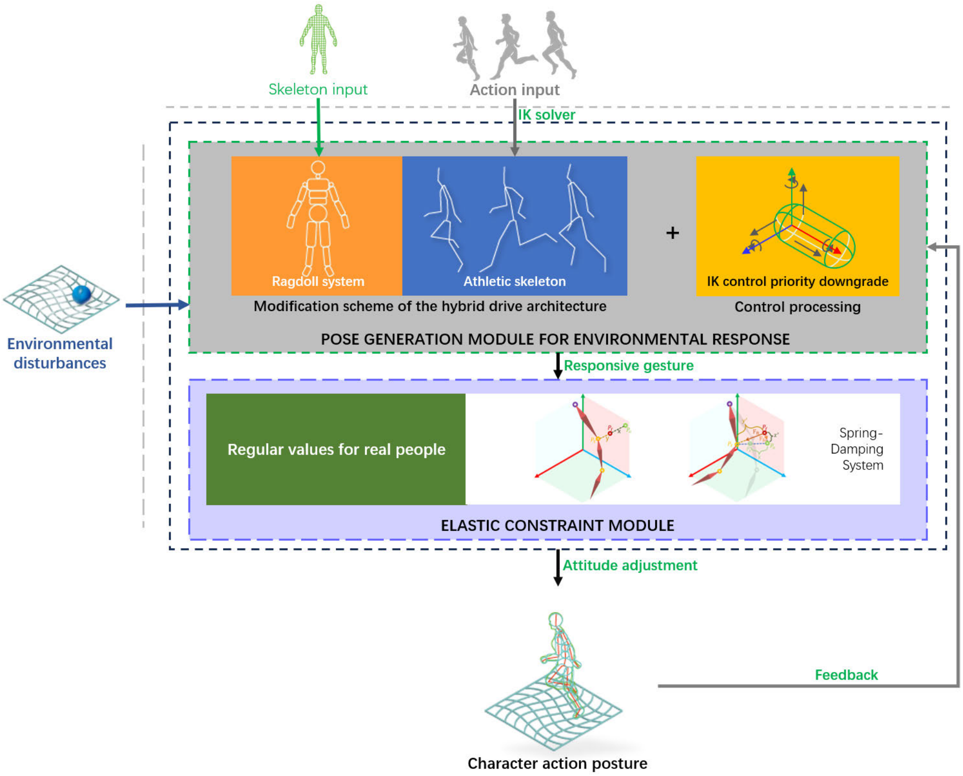

Skeleton input
Action input
IK solver
Ragdoll system
Athletic skeleton
+
IK control priority downgrade
Modification scheme of the hybrid drive architecture
Control processing
POSE GENERATION MODULE FOR ENVIRONMENTAL RESPONSE
Environmental disturbances
Responsive gesture
Regular values for real people
Spring-Damping System
ELASTIC CONSTRAINT MODULE
Attitude adjustment
Feedback
Character action posture

FIGURE 2. **Overview of our animation pipeline.**

the various approaches proposed in [38], this study employs the forward and backward reaching inverse kinematics (FABRIK) algorithm [47], which iteratively updates the body position data in both forward and backward directions to generate motions fast and efficiently. FABRIK has been widely applied in various scenarios [48], [49], [50], and its performance remains good even after modifications [47], [51], [52], [53], demonstrating strong scalability. This study combines the IK with physics-based characters and designs the FABRIK-ER (environment-responsive) algorithm, making the behavior generation independent from the force generation but subjected to forces, allowing the characters to interact with an environment and produce smooth gesture transformations.

C. POSTURE CHANGES OF HYBRID MODEL CHARACTER IN RESPONSE TO ENVIRONMENTAL CHANGES

The environmental influence represents the main external cause of posture changes of virtual humans and thus represents an important problem in character motion control. Recently, extensive efforts have been made to solve this problem. In [37], the PD controllers were used to train interactive fighting characters. In [54], the authors used an SPD controller to generate relatively stable character animations. In [31], sequential Monte Carlo (SMC) sampling was performed to synthesize actions. Han employed contact dynamics to generate different actions, such as walking, climbing, and dancing [55]. Peng used reinforcement learning to generate skill actions [33]; Hong performed the data-driven action prediction to realize ball control and shooting behaviors [36]. Lee adopted neural circulation networks to generate actions of interacting with a ball [56]. Posture change needs to address not only the problems of action data size and body optimization and the limitations and complexity of an algorithm but also extensive computing power consumption and numerous posture change anomalies generated by the excessive number of body bones.To this end, Lewin proposed a rod-constrained posture optimization scheme after reducing the number of bones [57].Carensac designed a touchdown stabilizer for the legs to compensate for the standing control feedback for the legs [58].However, the training effectiveness of using artificial intelligence to develop interactive methods is highly dependent on data, consumes a large amount of computational power, and often results in unnatural movements [31], [33], [37], [55], [56], [57]. Additionally, for actions outside the target task, further training is required. Methods like [36], which combine IK with physics-based characters, can only demonstrate good results on flat terrain if the interaction with the environment is not well-handled, making them unsuitable for complex environments.The ground contact stabilizer referenced in literature [58]is incapable of making corrections when a substantial force is exerted at low frequencies. The performance of the controller is contingent upon the selection of the starting point, and it demonstrates a lack of robustness under low frequencies.Reference [54]used the SPD controller to improve the stability of the PD controller to some extent, but forward dynamics pose significant challenges for achieving end-to-end tasks. In this study, the FABRIK-ER algorithm is designed, lightweight physics is used on the torso to construct the physical simulation structure of the body to reduce the computing power, and the under-fitting problems when the IK algorithms are used for complex body control are avoided. Finally, based on the properties of the hybrid framework combining the physics and IK, a specialized posture-based humanoid constraint method called the Elastic Constraint method is designed. This method applies reasonable directional attraction control to the joint motion changes, allowing the elbow and knee joints to adjust automatically to more reasonable positions and angles, thus optimizing the motion postures.

III. METHODS

This paper presents a hybrid structure designed for humanoid characters, integrating the lightweight physics-based body with the kinematic model-controlled limbs. This structure does not require using artificial intelligence methods and can generate motion in real-time with only a small amount of motion data. In addition, it can respond to the environment, as shown in Fig. 2.

The proposed solution first designs a pose generation module for environmental response and an elastic constraint module for a physics-based character. The core algorithm of the pose generation module for an environmental response, the FABRIK-ER algorithm, can heuristically solve the existing control data to generate action data that responds to the environment, thus simply and efficiently solving the problems of action control and environmental response. Structurally, the environmental response posture generation module represents a combination of a hybrid driving structure and a control processing module. The hybrid driving framework is based on a physics-based character, that is, the character model of a traditional rag-doll system. By combining this framework with kinematics, a hybrid driving structure is obtained. However, this structure uses IK to solve control data to obtain character actions and uses the physical body to respond to the environment. These two are performed in parallel to achieve realistic action effects for a character. This structure can achieve the function of controlling character actions and environmental responses simply and efficiently. For the purpose of control, an IK control priority downgrade strategy is designed to address command conflicts caused by different physical and kinematic control attributes in the hybrid driving structure. By making environmental response the priority in the solving process, some of the penetration problems caused by command conflicts are resolved. Next, to make the physics-based character's actions more human-like, an elastic constraint module is designed using regular human action rules to optimize a character's actions. It is needed only to control certain parts of the body to calculate a reasonable full-body posture. This solution mainly adopts a spring-damping system to solve the pole position in IK, affecting the calculation results of the body posture, making the posture of the middle limbs of the body more in line with human physiology visually, and improving the human-like effect of a character's actions.

As shown in Fig. 1,the blue and purple character model in the scene denote virtual characters rendered from real human motion data, and the white and orange character model are virtual characters rendered using the proposed model. According to the experimental results, the blue and purple character model ignores the changes of the environment and experiences a penetration phenomenon during walking.However, our virtual human can constantly adjust its foot movement to adapt to the rugged terrain due to the introduction of the pose generation module for environmental response. Moreover, due to the introduction of the elastic constraint module, the movement is more smooth and humanoid.

A. MODIFICATION SCHEME OF HYBRID DRIVE ARCHITECTURE

The proposed hybrid drive structure is shown in Fig. 3. The character model is divided into limb end parts and non-limb end parts, and the initial physical effects of the body are realized when rigid bodies and interaction elements are added to the two types of parts, as shown in Fig. 3(A). The location data on the limb end parts are used as a desired location, and the rigid body components and interaction elements of the limb end parts are moved to the desired location. The environment interaction components of the limb end parts are set at the desired location, and a rigid body with a negligible weight is configured to reduce the physical properties to prevent body flutter, as shown in Fig. 3(B).

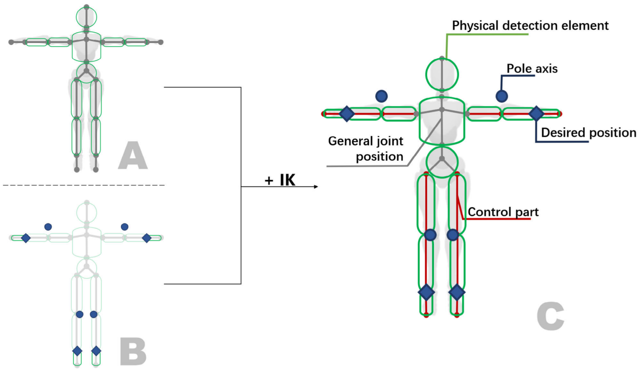


FIGURE 3. The modification scheme of the hybrid drive architecture: (A) the rag-doll system component configuration; (B) IK preset component settings; (C) the hybrid drive architecture.

Next, the IK controller is improved by placing four position points at the front and back of the limbs of the humanoid (bipedal) model as elbow and knee orientation points to adjust the overall orientation of the limbs. Subsequently, the IK controller is added to the end part of the limbs with a control chain length of two. The controlled joints reach the root of the limbs. By using the desired position as a target position control point and the orientation as an elbow and knee orientation point, the final effect is achieved, as shown in Fig. 3(C). Furthermore, to ensure the tightness of a character's body, the joint level of the model takes the crotch as a root and adds joint articulation parts for the non-extremity end parts, extremity end parts, and target position control points so that the target position control points fit the extremity end parts, thus preventing the target position control points from scattering in the environment.

B. POSE GENERATION MODULE FOR ENVIRONMENTAL RESPONSE AND AN ELASTIC CONSTRAINT MODULE

The IK control of the body is independent of a lightweight physics body, and their combination can provide only a slight response to the environment. Therefore, this study proposes the FABRIK-ER method based on the FABRIK concept and designs a hybrid-driven structure for the IK controller algorithm. This allows all the physically controlled parts under IK to respond to the environment. In the iterative process, the desired location represents the desired position of the end joint, and the calculation process aims to make the end effector as close as possible to the desired position. The pole axis represents the orientation point for adjusting the position of the intermediate joint during the iteration. The overall motion of the joint points toward the pole, as shown in Fig. 4.

This section introduces the methods for calculating and adjusting positions, the torque algorithm, and joint control processing in the FABRIK-ER algorithm.

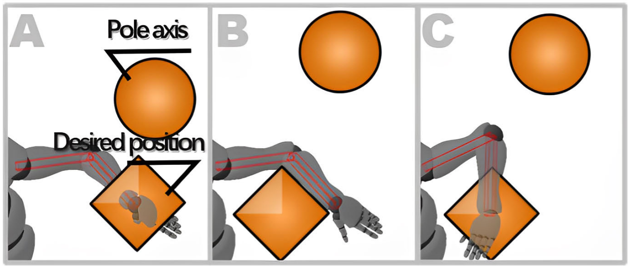


FIGURE 4. Illustration of the FABRIK controlling the effect of the character's left upper limb: (A) the desired position and pole axis are not modified; (B) the desired position and pole axis position are modified; (C) the joint changes after modification.

1) CALCULATION OF THE POSITION OF THE END JOINT TO THE ROOT JOINT

The animation pipeline used in this study is updated in real time, and when an attitude change occurs, the IK-controlled component will first solve for the position of the control joints according to the target position; the solution process is shown in Fig. 5; Eq.(1) provides the calculation equation for each joint position from the ends of the limbs to the trunk of the body.

$$\begin{cases} P'_n = P_T, n = 0 \\ P'_n = \dfrac{\overrightarrow{P'_{n-1}P_n}}{\left|\overrightarrow{P'_{n-1}P_n}\right|} * l_{n-1} + P'_{n-1}, n \geq 1. \end{cases} \tag{1}$$

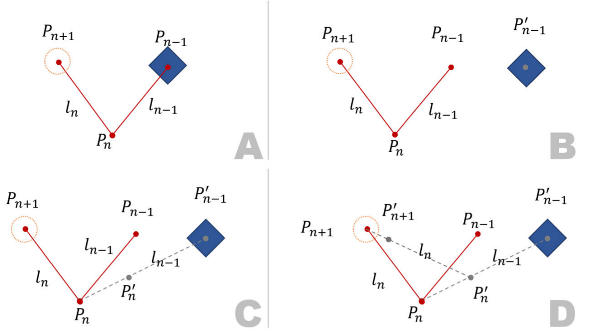


FIGURE 5. Procedure for calculating the new positions of all joints, from the end joints to the root joints: (A) there is no change in the target point; (B) target points begin to change; (C) position P'_n calculation; (D) position P'_{n+1} calculation.

In Eq. (1), n is the number of joints controlled by IK; P'_n denotes the new position; P_n is the original position; when $n = 0$, $P'_0 = P_T$, where P_T is the position of the target position control point; l_{n-1} is the length of the part of $\left|\overrightarrow{P_{n-1}P_n}\right|$; P_n is the parent joint point of P_{n-1}.

The calculation of joint positions is performed from the limb endpoints to the torso. When $n = 0$, P_0 is the end position, which is consistent with the desired position P_T; when $n \geq 1$, the parent joint point P_n is the starting position; P'_n is calculated from the known values of P'_{n-1} and P_n until all new joint position data are obtained.

2) INTERMEDIATE OVERALL JOINT POSITION ADJUSTMENT

After the calculation by Eq. (1), the preliminary position data of the joint are obtained. These position data are substituted into Eq. (2), from which the pole axis is used to obtain the final position data. The calculation process is shown in Fig. 6, and the specific mathematical expression is as follows:

$$P'_n = \theta_{\overrightarrow{P_{n-1}P_{n+1}}} * \overrightarrow{P_{n-1}P_n} + P_{n-1}, M > n > 0, \tag{2}$$

where M is the number of IK-controlled parts; P'_n is the calculated new position; $\theta_{\overrightarrow{P_{n-1}P_{n+1}}}$ is the angle of rotation with $\overrightarrow{P_{n-1}P_{n+1}}$ as an axis; $\overrightarrow{P_{n-1}P_n}$ is the vector of P_{n-1} pointing to P_n; P_n is the intermediate point of points P_{n-1} and P_{n+1}; P_t is the original pole axis, and P'_t is the changed pole axis.

When adjusting the middle joint, a plane passing through P_{n-1} is generated for the normal plane from P_{n-1} to P_{n+1} and point P_t with an unchanged middle joint position and changed point P'_t are mapped to this plane; in addition, angle θ between the two mapped points and point P_{n-1} is calculated. Finally, $\overrightarrow{P_{n-1}P_n}$ is allowed to go around $\overrightarrow{P_{n-1}P_{n+1}}$ as the axis rotates by θ degrees.

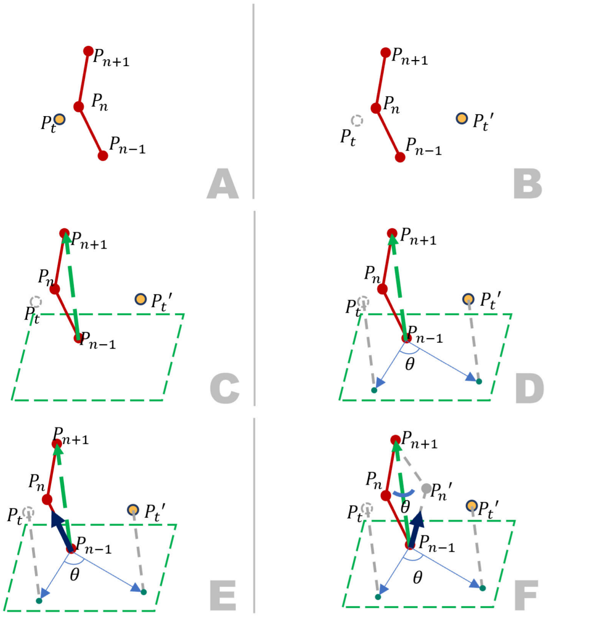


FIGURE 6. **Illustration of the calculation process of the adjustment of the middle overall joint position: (A) pole axis before moving; (B) pole axis while moving; (C) a plane is perpendicular to point P_{n-1} pointing to point P_{n+1} vector through point P_{n-1}; (D) points P_t and P_t' are mapped onto the plane to generate angle θ with point P_{n-1}; (E) point P_{n-1} points to point P_n as a vector; (F) position of point P_n given as a rotation of the vector by angle θ.**

3) ASSIGNING VALUES FROM IK ROOT JOINT TO CHILD JOINTS

Equations (1) and (2)calculate positions based on different components, and their respective impacts are shown in Fig. 7. Directly assigning the calculated final position data to the corresponding body joints can result in a loosely connected and disordered skeletal structure for the humanoid (bipedal) model. This is because the results of Eq. (1) might cause changes in the root joint position depicted in Fig. 7(A), which can affect the control position data for iteration, leading to unreasonable body pose calculations. However, human motion is generated by the skeleton rotating around joints, and directly solving the joint position offsets is not reasonable. Therefore, the position transformation data of other joints cannot be directly applied to the humanoid (bipedal) model. Instead, in this study, the torque quaternion is obtained by Eq. (3), and the torque angle is obtained by Eq. (4). The torque is applied from the root joint to the child joints, which can be expressed as follows:

$$a_n = q_n^{-1} * q_r, M > n >= 0, \tag{3}$$

in Eq. (3), M is the number of IK-controlled parts; n is the number of IK-controlled joints; a_n is the quaternion used for rotation of the corresponding joint; q_n^{-1} is the inverse quaternion of the corresponding joint; q_r is the root joint.

$$\partial = \overrightarrow{P'_n P'_{n+1}} * q_{a_n}^{-1}, M > n >= 0. \tag{4}$$

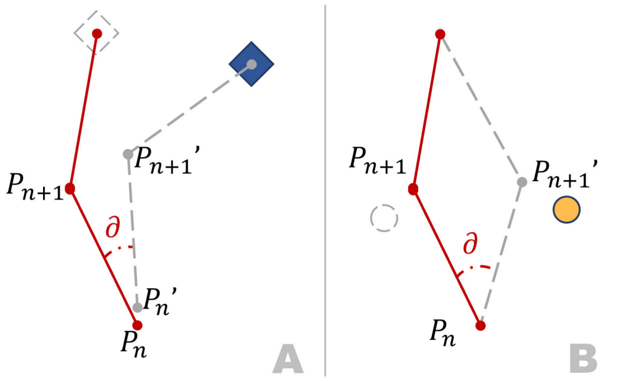


FIGURE 7. **Assignment of IK root joints to sub-joints. (A) Only the end controller is moved; (B) only the pole axis is moved.**

In Eq. (4), P'_n and P'_{n+1} are the corresponding joint position data calculated by Eqs. (1) and (2), respectively; $q_{a_n}^{-1}$ is the inverse quaternion calculated by Eq.(3), ∂ is the current rotation Angle of the joint with respect to the root joint.

Equations (3) and (4) start from the root joint, calculate the rotation angle of each joint to its sub-joint by inverting the quaternion, and let the root joint rotate the angle ∂. Next, the position of the sub-joint is calculated based on the length of the joint and the step-by-step recursion until the rotation angle and position data of all joints are obtained.

4) IK CONTROL PRIORITY DOWNGRADE STRATEGY

According to Eq. (1-4), the position and rotation angles of the entire joints of character's are controlled by the IK root joint, the expected position, and the position data of the pole axis. The position of the root joint is determined by itself, and the rotation angle of the root joint is obtained based on the position of the sub-joint. The position and rotation angle of the end joint are determined by the desired position. In common 3D engines, both the skin and the rigid body have their own positions and rotation angles, and the computational priority of the rotation and offset of the skin mesh is higher than that of the rigid body. The rigid body responds to the environment according to the skin mesh data, while IK controls skin data and does not process rigid-body data. As a result, the proposed hybrid drive structure is not completely affected by an environment; for example, there is unreasonable distortion of some limbs when they interact with the environment, as shown in Fig. 8.

As shown in Fig. 9, when an interactive object is dragged upward from the foot, the foot responds by lifting and contracting, and the position and distance length of each joint remain unchanged and rotate to a suitable angle so that the overall joint can provide a good response to environmental changes. When an interactive object passes through the middle joint, the strong constraint of IK on the skin leads to the phenomenon of lower limb penetration. Further, when an interactive object moves horizontally and touches the IK root part, since the rotation of the IK root part is determined by the position of the sub-joint, and the position of the IK root joint is

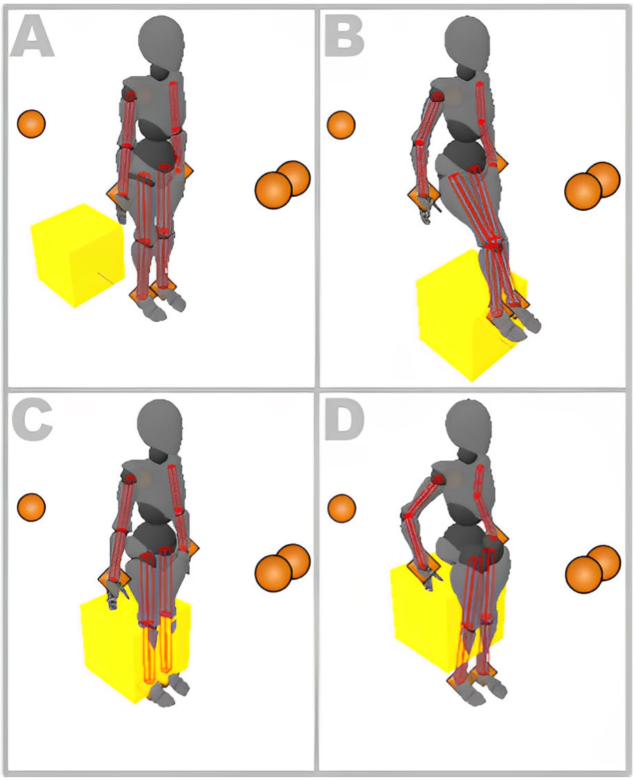


FIGURE 8. Effects of interactive objects touching different parts of the lower limbs: (A) lower limbs without collision; (B) collided feet; (C) lower legs collided abnormally; (D) thighs collided abnormally.

not affected by IK, the IK root part will not be able to respond to environmental changes by rotation but by pulling offset. In this case, the change in the sub-joint's position makes the IK root joint rotate at a new angle. At the same time, since the root joint is the crotch, and the body joint hierarchy defines that the parent joint affects its sub-joint, the change of the IK root joint's position does not affect the crotch, which results in the stretching deformation of the crotch and thigh.

Therefore, an improved method is proposed, and a response pose correction module is designed to solve the phenomenon of model penetration in interactive motion and modify the response pose. The control priority of IK in the skin offset rotation is higher than that of interactive detection elements, which causes weakened responsiveness of a character's environment and unreasonable stretching of the body. This study addresses this problem by lowering the priority of skinning control. The IK motion control is modified as follows:

$$P_e, \partial_e \longrightarrow P_r, \partial_r, \tag{5}$$

where P_e and ∂_e are the offset position and rotation angle of the mask, respectively; P_r and ∂_r are the offset position and rotation angle of the rigid body, respectively.

By modifying the control object of IK from skin to the rigid body, the priority of the rigid body's detection response to the environment is made higher than that of the IK's driving control of the rigid body so that the character's body can give priority to the environment during motion control without causing problems of partial piercing and stretching deformation.

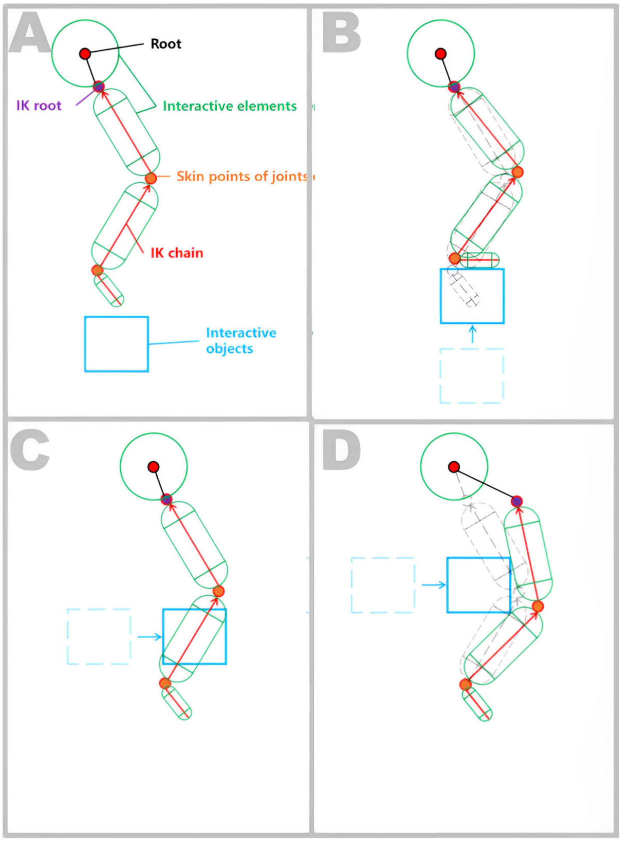


FIGURE 9. Illustration of interactive objects touching different parts of the lower limb: (A) lower limb without collision; (B) collided foot; (C) lower leg collided abnormally; (D) thigh collided abnormally.

C. ELASTIC RESTRAINT ADJUSTMENT OF BODY POSTURE

In the proposed pose generation module, the posture of limbs depends on the root and endpoint of each limb, as well as on the pole axis, as shown in Fig. 10. However, in general, the pole axis is a stationary point, which results in unnatural body movements for a character and even violates the torque of human joints, as shown in Fig. 11. Therefore, to achieve natural movements, this study first imposes joint constraints on the range of motion of the lightweight physical body. Then, a spring-damping system is introduced, and an elastic constraint method is designed based on the typical orientations of human movements. This improves the humanoid nature of a character's movements. A detailed explanation of these techniques is presented in the following.

1) SPRING-DAMPING SYSTEM

The spring-damping system represents a conventional vibration system, which can generate smoother trajectories through damping compared to a spring system. Based on this principle, this study designs an anti-vibration elastic

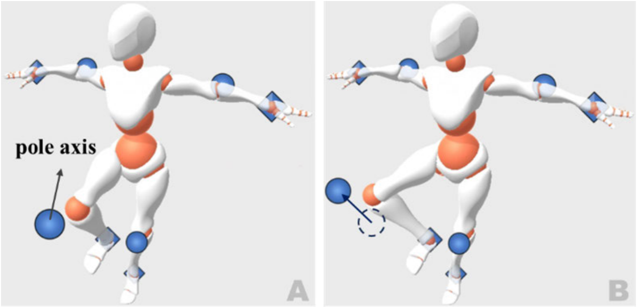


FIGURE 10. Joint orientation of the humanoid (bipedal) model controlled by the pole axis: (A) the effect of the pole axis before moving; (B) the effect of the pole axis after moving.

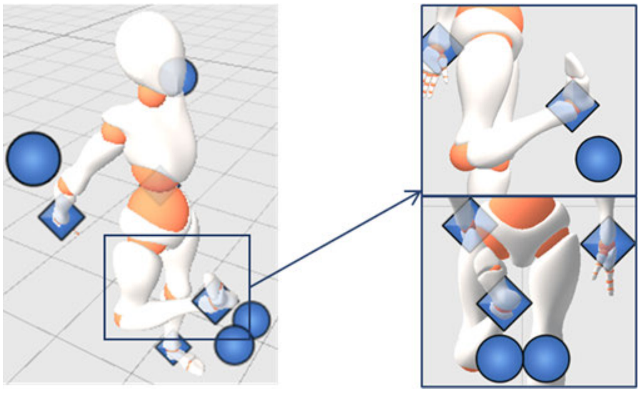


FIGURE 11. Abnormal human knee abduction due to the pole axis control.

constraint control system for the pole axis. In this system, the spring serves as a solution for initiating offset positions, while the damping serves as a shock absorption solution. The specific method is defined as follows:

$$F = -kx - bv, \tag{6}$$

where F is the action force, k is the spring tightness parameter, b is the damping factor, and v is the relative velocity between the spring connections.

The constraint used in this study is different from conventional action constraints. For example, conventional constraints aim to optimize the input data by limiting the movement amplitude so that the torque level does not violate the regular laws of real human movement. However, the elastic constraints used in this paper are based on the regular movement patterns of real people, and the spring-damper system is used to stretch the pole axis, which allows a character to make more action postures and conform to the regular movement laws of real people. A spring-damping system is selected to pull the pole axis. The spring is responsible for stretching the pole axis, and the damper is introduced to reduce the vibration generated by the spring when stretching the pole axis so that the pole axis can rapidly return to a stable state during the deflection process.

2) ELASTIC CONSTRAINT MODULE

Generally, constraints define the range of a joint's degrees of freedom in a physical humanoid (bipedal) model. Under the control of extreme movements, a constraint can restrict the movement pose of a humanoid (bipedal) model, while in IK, the pole axis controls the overall orientation of the middle joint, and no constraints are applied. In such a case, it is needed to control the pole axis so that it can move within a reasonable range to achieve the constrained state [20].

To make the human action posture present a reasonable torque in most cases, this study controls the movement of the pole point, which in the case of a human-like (bipedal) model is an elastic constraint on the torque of the body limbs. In this way, the pole point can move within a reasonable range as much as possible.

The basic principle of elastic restraint is that the pole axis is spring stretched by the constraint point P_o outside the body and the position point P_t of the elbow and knee joint to make the pole axis move within a reasonable range. The specific algorithm for this is shown in Fig. 12, where P_o is the constraint position point that always follows the conventional data of the elbow and knee; P_f is the pole axis; P_t is the elbow and knee point; x and y are the positional distances of P_o from P_f and of P_f from P_t before offsetting, respectively, as shown in Fig. 12(A); x' and y' are the positional distances of P_o from P_f and of P_f from P_t after offsetting, respectively; P_{ft} and P_{fo} are the forces that stretch P_t and P_o by P_f, respectively; and P_x is the position point where P_o moves towards P_t, and its distance is x, as shown in Fig. 12(B).

First, according to Eq. (6), the calculation process of the spring-damped system requires obtaining the current velocity of the pole axis, which is given by:

$$v_f = \frac{\mathrm{d}\left|\overrightarrow{P_f}\right|}{\mathrm{d}t}, \tag{7}$$

where $\left|\overrightarrow{P_f}\right|$ is the position distance after the position change, t denotes the time, and v_f is the current velocity of P_f.

In the process of moving, it is necessary to calculate the velocity of the current point P_f in real time, which is then used to calculate the tensile force of the constraint point P_o outside the body and the position point P_t of the elbow and knee joint on the pole axis. The calculation equation of the force is shown in Eqs. (8) and (9).

$$F_{ft} = -k_{ft}(y' - y) - b_{ft}v_f. \tag{8}$$

In Eq. (8), F_{ft} represents the stretching force exerted by P_t on P_f; k_{ft} is the elastic coefficient for stretching P_t and P_f; y is the distance between P_t and P_f before movement; y' is the distance between P_t and P_f after movement; b_{ft} is the damping coefficient for stretching P_t and P_f; v_f represents the current velocity of P_f.

$$F_{fo} = -k_{fo}(x' - x) - b_{fo}v_f. \tag{9}$$

In Eq. (9), F_{fo} represents the stretching force exerted by P_o on P_f ; k_{fo} is the elastic coefficient for stretching P_o to P_f; x is the distance between P_o and P_f before movement;x' is the distance between P_o and P_f after movement; b_{fo} is the

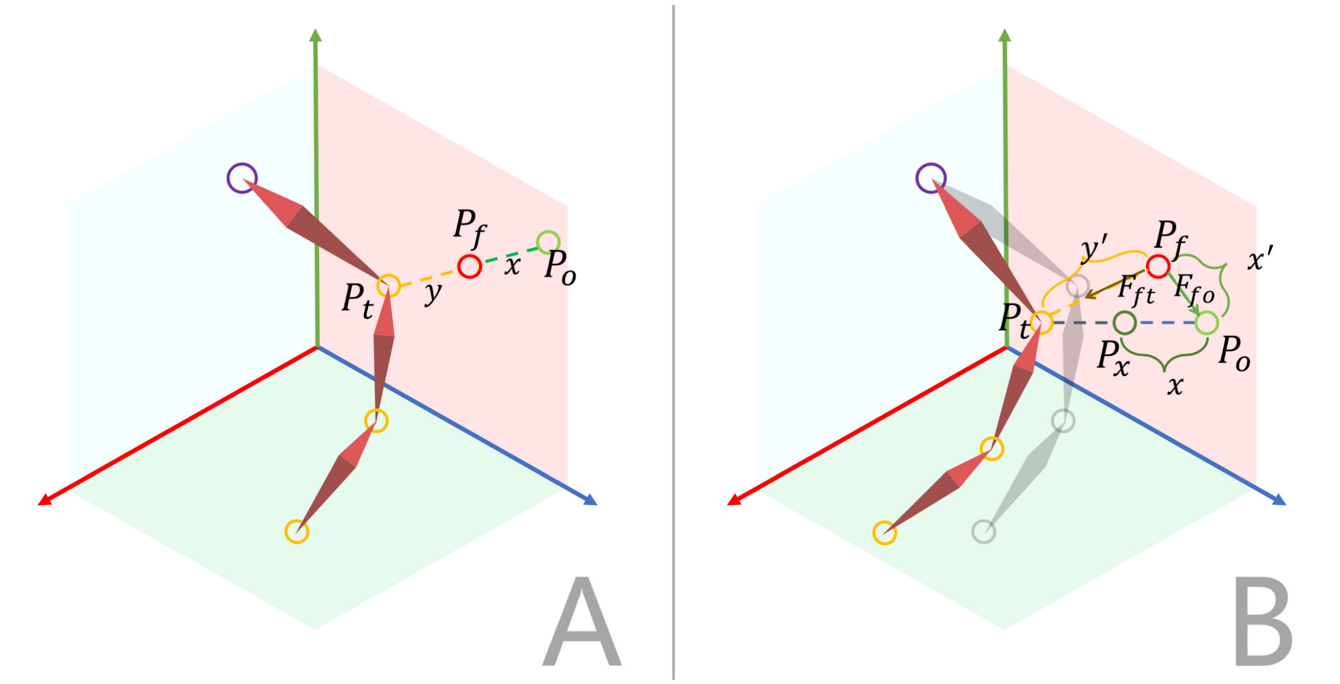


FIGURE 12. The change in the elastic constraint algorithm when IK controls motion: (A) IK control joints are not moving; (B) IK control joints are moving.

damping coefficient for stretching P_o and P_f; v_f represents the current velocity of P_f.

After obtaining the two stretching forces of P_o and P_t on P_f, the resultant force needs to be calculated to determine the direction of movement. The calculation process is defined by Eqs. (10)- (15). To calculate the resultant force, the two forces need to be decomposed into two perpendicular forces. The first step is to calculate the angle between the two forces. This angle is used to determine the relationship between the two forces, and the equation for calculating the angle is as follows:

$$\theta = arc\frac{\left|\overrightarrow{P_fP_o}\right|^2+\left|\overrightarrow{P_fP_x}\right|^2-\left|\overrightarrow{P_xP_o}\right|^2}{2\left|\overrightarrow{P_fP_o}\right|\left|\overrightarrow{P_fP_x}\right|}+ arc\frac{\left|\overrightarrow{P_fP_t}\right|^2+\left|\overrightarrow{P_fP_x}\right|^2-\left|\overrightarrow{P_xP_t}\right|^2}{2\left|\overrightarrow{P_fP_t}\right|\left|\overrightarrow{P_fP_x}\right|}, \tag{10}$$

where θ represents the angle between forces F_{ft} and F_{fo}.

The direction of F_{ft} is taken as a positive direction. When $\theta > 90°$, force F_{fo} weakens force F_{ft}. The force in the positive direction and the force perpendicular to the positive direction can be calculated by:

$$\begin{cases} F_{pd} = F_{ft} - \cos\left(180° - \theta\right) F_{fo} \\ F_v = \sin\left(180° - \theta\right) F_{fo} \end{cases}, \theta > 90°, \tag{11}$$

where F_{pd} represents the force in the positive direction, which is perpendicular to F_{ft}.

When $\theta < 90°$, force F_{fo} enhances the force F_{ft}. The force in the positive direction and the force perpendicular to the positive direction can be calculated by:

$$\begin{cases} F_{pd} = F_{ft} + \cos\theta F_{fo} \\ F_v = \sin\theta F_{fo} \end{cases}, \theta < 90°, \tag{12}$$

where F_{pd} represents the force in the positive direction, θ is the angle between forces F_{ft} and F_{fo}, F_{fo} is the stretching force of P_o on P_f and F_v represents the force perpendicular to F_{ft}.

When $\theta = 90°$, forces F_{fo} and F_{ft} are perpendicular to each other. The force in the positive direction and the force perpendicular to the positive direction can be calculated by:

$$\begin{cases} F_{pd} = F_{ft} \\ F_v = F_{fo} \end{cases} \quad \theta = 90°, \tag{13}$$

after the two perpendicular forces are obtained by Eqs. (10)-(13), the resultant force can be calculated using the Pythagorean theorem as follows:

$$F_{sum} = \sqrt{F_{pd}^2 + F_v^2}, \tag{14}$$

where F_{sum} represents the magnitude of the resultant force of F_{pd} and F_v.

After calculating the resultant force's magnitude, its direction can also be determined by:

$$\overrightarrow{F_{sum}} = \frac{\overrightarrow{F_{pd}} * F_{\text{sum}}}{F_{pd}}, \tag{15}$$

where $\overrightarrow{F_{sum}}$ represents the direction of pole movement, F_{pd} is the force in the positive direction, $\overrightarrow{F_{pd}}$ denotes the direction in the 3D space, and F_{sum} is the resultant force of F_{pd} and F_v.

After performing the calculations described above, the final result is the movement direction of the pole axis.

IV. RESULTS AND DISCUSSION

The proposed hybrid driving framework and optimization scheme developed to achieve a more reasonable human-like environment response were verified by experiments, and the experimental results are presented in the following.

A. EXPERIMENTAL DESIGN

1)Equipment setup: This study used the 2019.4.17 version of the Unity 3D engine as an editing engine, Visual Studio as a programming environment, and Windows 10 as a cross-platform operational system. The *C#* scripts were used, and the computer configuration included an Intel(R) Core (TM) i7- 9700 CPU @ 3.00 GHz processor, Intel (R) UHD Graphics 630 and NVIDIA Ge Force RTX 2070 graphics cards, 16.0 GB of RAM, and a 64-bit operational system.

2)Lightweight physical avatar construction: In Unity, a lightweight physics-based character with different masses assigned to different body parts was constructed. For the interaction with the environment, collision detection was used to handle it. The body parts were divided into simple geometries for easier processing.

A conventional, physical, and virtual human needs to have a rigid body to obtain physical properties to be able to receive forces and generate motion effects in the Cartesian space. Rigid body data on different parts of the human body are different. In [59], the authors studied the data on various parts of the human body, and a virtual human used in this study refers to that presented in [59] regarding the rigid body data on different body parts, as shown in Table 1.

TABLE 1. Weights of different body parts (%).

Body Part	Weight Ratio
head	5.78%
neck	1.61%
upper arms	3.17%
lower arms	2.18%
hands	0.77%
upper trunk	44.08%
lower trunk	24.80%
thighs	10.06%
calves	6.04%
feet	1.50%

Physical elements were required to have a physical avatar, and interaction detection represented a basis for judging whether the rigid body of the avatar could accept the force. Since the skin of the virtual human has a complex geometric shape, it could be challenging to construct the outer frame of complex detection elements and combine them with IK for hybrid implementation. Therefore, this paper adopted the geometric elements of lightweight physics for interaction detection and used data in Table 1 to construct a virtual human, as shown in Fig. 13(A). Since capsules closely resemble the shape of the human body, this study mainly used capsules as geometric interaction elements for the entire virtual character, except for a few exceptions (e.g., the head), which were represented by a sphere, as shown in Fig. 13(B).

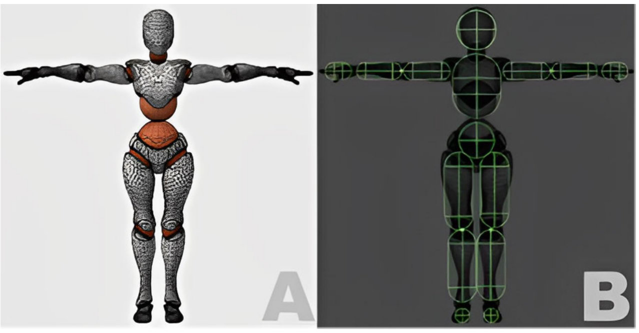


FIGURE 13. A lightweight, physical, virtual avatar: (A) virtual human skin and its mesh; (B) simple geometric interaction elements of a virtual human.

When a human model sets the skin weights for skeleton binding, the generated skeleton layers represent a logical framework of forward kinematics (FK), and the lower layers inherit the information from the upper layers. The human torso can be divided into three main parts: head and neck, torso, and limbs; the torso is an important hub connecting the head, neck, and limbs. The crotch in the torso is the core of the human body, so when generating a skeleton, the skeleton level of a virtual human model generally uses the crotch as a root to construct joints.

The body joint hierarchy of a virtual human typically sets the crotch as a root joint and connects it outward to implicate the whole body. In this study, the conventional skeleton hierarchy was used for joint construction for the purpose of posture control of a physical, virtual human. The adopted construction hierarchy is shown in Fig. 14, where it can be seen that the crotch was the root as a pivot; the extremities and the head denoted the ends to receive the upper data information, and the key information was placed on the limbs.

B. LOCAL MOTION CONTROL

In this study, localized motion control of the virtual human was performed by inputting keyframe animations of limb contractions and extensions for a single body part. The upper limb movements lasted 1.6 s, the lower limb movements lasted 1.2 s, and a frame was extracted every three frames (i.e., one frame was extracted every 0.1 s), as presented in Figs. 15and 16. The purpose of this experiment was to test the effects of the elastic constraint and the posture correction

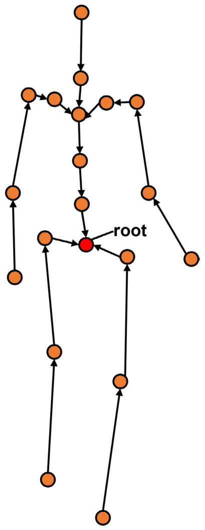


FIGURE 14. Joint hierarchy of general virtual human.

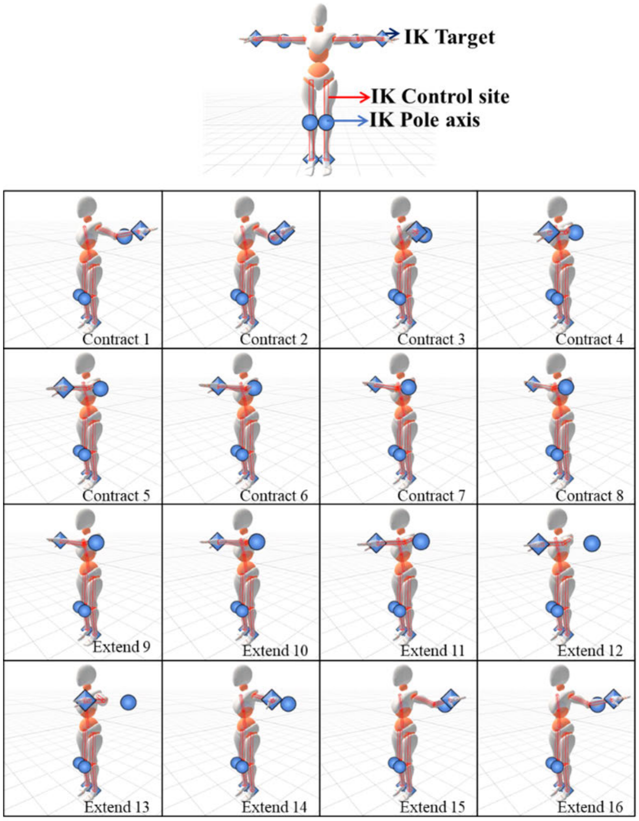


FIGURE 15. Upper limb contraction and extension exercises.

module on the motion control performance. To observe the effect directly, this study adopted a single variable control method, where only a single part (e.g., a hand or foot) was used in the keyframe animation.

Upper limb contractions and extensions are shown in Fig. 15, where it can be seen that the elastically constrained pole axis shifted its position throughout the contraction and extension. The elbow orientation was slightly downward during the contraction but slightly upward during the extension, which was due to the offset of the pole axis, fitting the real person's movement. During the hand contraction and extension processes, even if only the position was controlled, the hand was rotated and adjusted under the pulling of the elbow so that it did not maintain the same position all the time.

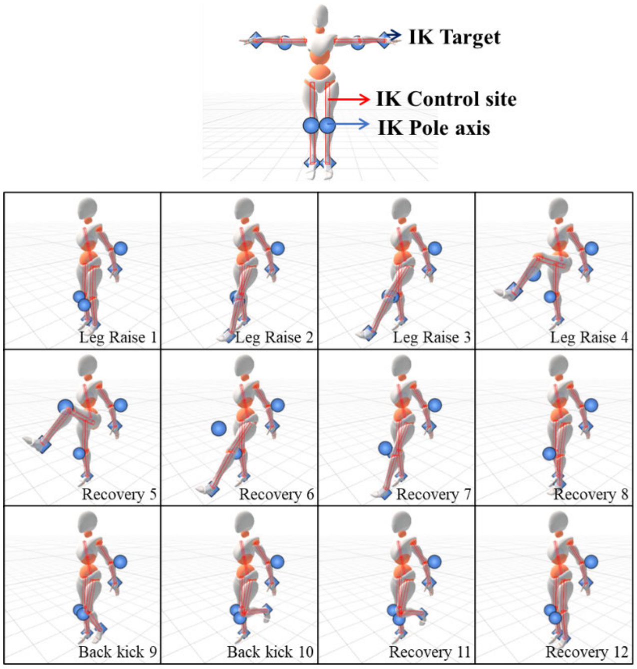


FIGURE 16. Lower limb lift and back kick exercise.

As shown in Fig. 16, the pole axis was constantly shifted throughout the movement, the foot was always moving in the same direction as the key-frame animation of the position control, and there was no anti-joint orientation (refer to the video for more details).

In this experiment, the model did not interact with the environment and was solely controlled for movement. Under the influence of elastic constraints, it was able to compute actions that closely resemble human behavior. The length of the IK-controlled segments remained consistent with the original length of the body parts, avoiding any stretching or deformation. This demonstrates that the posture correction module effectively enables the character's movements to exhibit a high degree of human-like qualities and allows for flexible control of body orientation.

The elastic constraint method proposed by us can enable the character to naturally generate reasonable actions. In order to further verify this performance, we took the elastic constraint method as the baseline experiment, deleted the core function of the character's elastic constraint, and only retained the basic physical constraint, so that the character repeated the actions in Fig. 15 and 16. The experimental results are shown in Fig. 17 and 18.

As shown in Fig. 17, since the contractions and stretching exercise of the character's upper limbs are not very

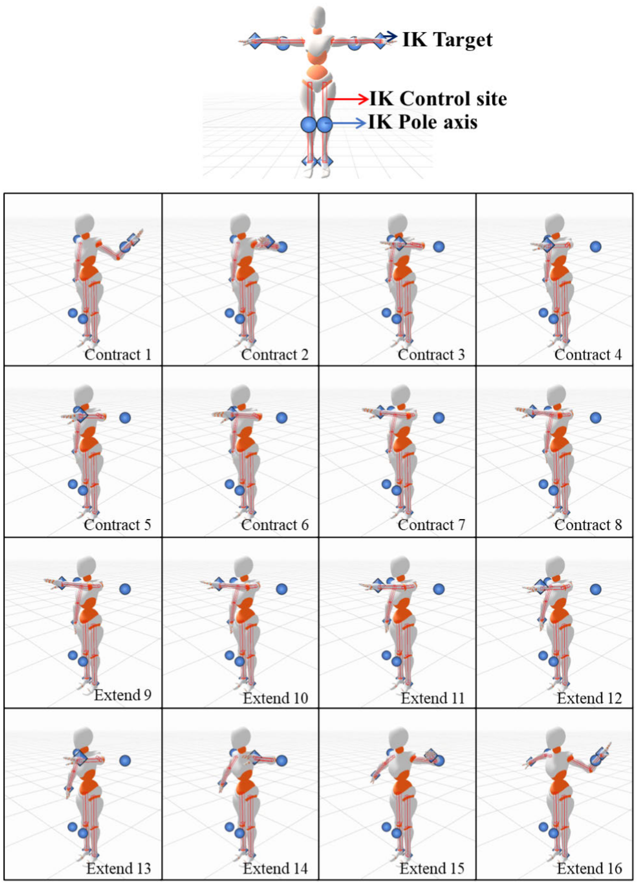


FIGURE 17. Upper limb contraction and extension exercise with basic constraints retained.

exaggerated, there is no abnormal posture. But except for the first and last hand movements, which are natural, the rest of the movements are very stiff and too mechanical.

Compared to the upper limb movements, the lower limb movements shown in Fig. 18 better illustrate that without the use of elastic restraints, the character generated movements are unnatural, most notably the anti-joint posture that occurs when the character lifts his feet. It should be noted that in the video recorded in the experiment, since we still retain the basic physical constraints, when the lower limb anti-joint occurs, it will automatically correct to a reasonable posture. However, due to the influence of IK, the physical constraints cannot be realized, so the shaking of the lower limb and even the whole body will be observed in the video. This also shows that our elastic constraints are very important, in addition to humanizing the character's actions, but also harmonizing the conflict between IK and the physical torso, the two complement each other, so that the character's motion-driven framework is more optimized.

C. REAL HUMAN ACTION DATA INPUT

To verify the rationality of the elastic constraints function, this study conducted experiments using real human data obtained from video motion capture. We extracted two-dimensional action data from a randomly selected video featuring real human actions. This data was then used to estimate three-dimensional action data, which was subsequently mapped onto the virtual character. To accomplish this, we devised a video action capture and data generation scheme, incorporating modules like HRNet and SemGCN. The centerpiece, HRNet, employs neural network models such as high-resolution networks and pose-time fusion networks. The architecture and operation of HRNet are depicted in Fig. 19.

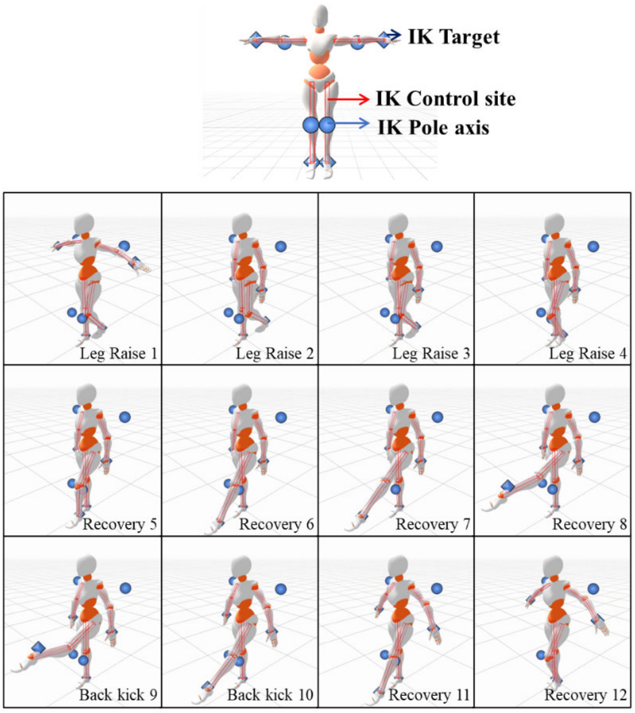


FIGURE 18. Lower limb lift and back kick exercise with basic constraints retained.

Since the acquired action data contained the offset data for each joint of the whole body, it is necessary to preprocess the data before they are input into the virtual character; that is, only the action data of the torso and the end of the limbs of the real person are extracted and input into the virtual character. For the action data, the walking action was selected as an input, and the whole action lasted 2.8 s; one frame was extracted every three frames (i.e., one frame was extracted every 0.1 s), as shown in Fig. 20. The virtual character represented by the real action data was denoted by A, which was only the model with bones, and the hybrid architecture virtual character modified by the proposed scheme was denoted by B. To observe the experimental results more scientifically, this study used and analyzed the data of the three axes of a virtual person from a medical image.

As depicted in Fig. 20, the virtual character demonstrated a convincing humanoid effect during the action operation, and the fitting effect was commendable, as showcased in the comparison screen of the live data input. Naturally, during the movement process, if the data input speed is too rapid, the character's action may slightly lag behind

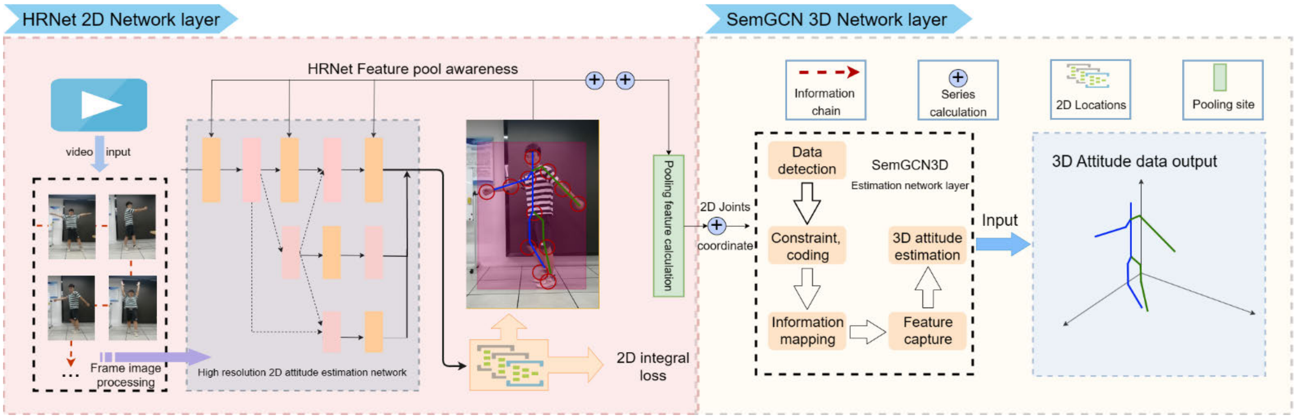


FIGURE 19. The scheme of video motion capture and data generation.

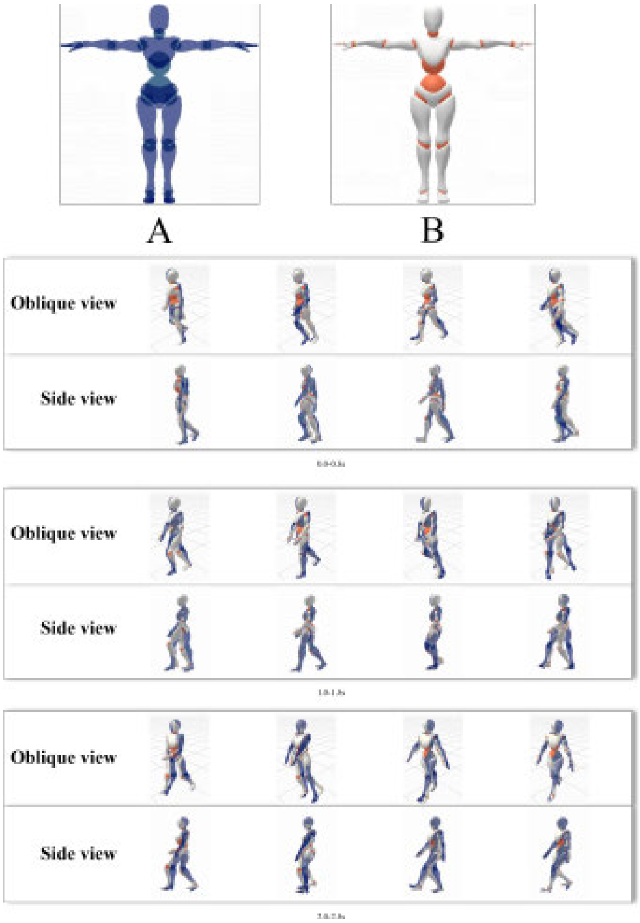


FIGURE 20. Comparison of oblique view and side view walking data.

the input action. However, once the movement ceases, the virtual human action, in conjunction with kinematic and physical simulations, eventually aligns with the real action data. The fact that controlling only a few parts can generate anthropomorphic actions for the entire body signifies that the elastic constraint method proposed in this paper effectively simulates the overall joint control of the virtual human.

This study focuses on the motion control of physically-based virtual humans, with simulation being one of its primary objectives. As illustrated in Fig. 21, the orientation of the elbow and knee under the elastic constraint remained well-fitted even in the absence of data input. Our virtual human actions ultimately align with real human action data, further affirming the high fidelity of our simulation scheme.

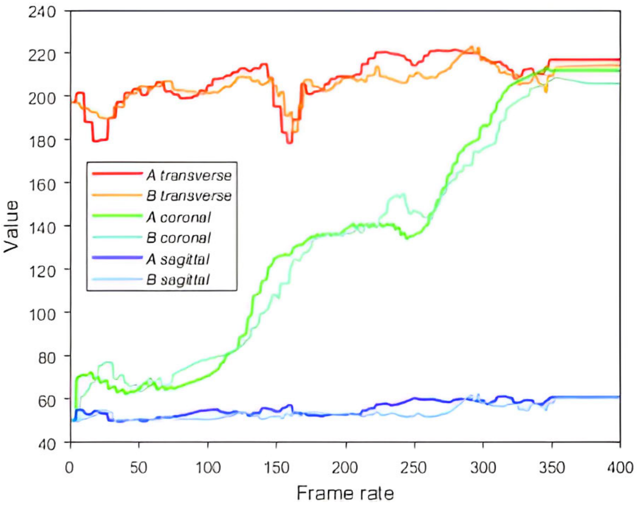


FIGURE 21. Variations in values of the transverse plane, coronal plane, and sagittal plane for models A and B walking on irregular terrain.

D. WALKING DATA INPUT IN IRREGULAR TERRAIN

A checkerboard patterned ground with a cell size of 28 × 28, which allowed direct visual observation of the ground surface undulations, was designed. The virtual character was made to walk on an irregular terrain to examine the effect of the model proposed in this paper on the interaction with the environment. In this experiment, the proposed virtual character used the aforementioned video motion capture technique to obtain the action data; the whole behavior lasted for 2.8 s, and one frame was drawn every six frames (i.e., one frame was drawn every 0.2 seconds), as presented in Fig. 22. In this experiment, the virtual character represented by the real action data was denoted by A, which was a blue and purple colored character model in the scene, and

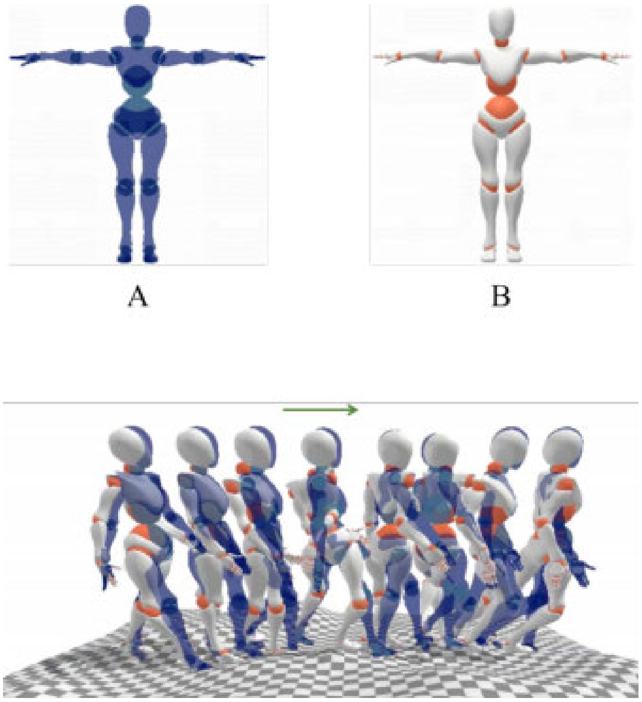


FIGURE 22. Effect of the model on the interaction with the environment for a virtual character walking on irregular terrain.

the hybrid architecture virtual character modified by the proposed scheme was denoted by B, which was a white and orange colored character model in the scene.

The action data presented in Fig. 22 were obtained from the video of a real person walking from left to right on a flat ground; it can be seen that model A disregarded changes in the environment, whereas model B continuously adjusted his foot movements to adapt to the rough terrain.

Next, to verify the response of the virtual character to the environment, an irregular terrain was selected randomly, and the actual data of the person walking were input into model A. The results are shown in Fig. 23, where the yellow character is model A, and the red character is B. During the walking process, character A's feet continuously penetrate the ground, whereas character B remains grounded. Throughout this process, the actions of both characters exhibit inconsistencies due to environmental responses and elastic constraints. This indicates that our virtual character B effectively interacts with the environment. During interactions, virtual character B automatically adjusts its posture in a manner consistent with natural human actions. Its action adjustments are self-regulated and not solely dependent on data sources, significantly enhancing the variety of character movements possible. This demonstrates that our character's action simulation is not confined to training datasets.

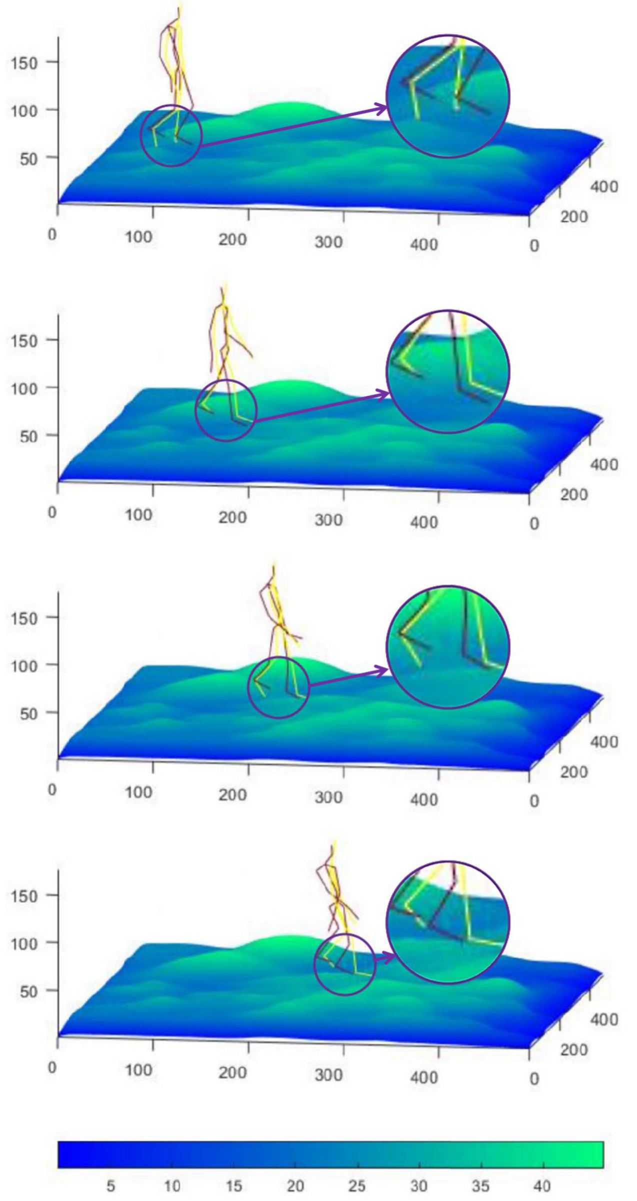


FIGURE 23. Data obtained from the video of a virtual character walking on irregular terrain; the yellow character represents model A, and the red character represents model B.

E. ENVIRONMENTAL RESPONSE EFFECT

To verify the response performance of the proposed virtual character to an environment under the control of the posture correction module further, a small ball with the speed, size, and weight values of one and a large ball with the speed, size, and weight values of two were struck at different parts of the virtual character's body. The cell size of the ground checkerboard pattern was also 28 × 28. In this experiment, the virtual character was standing on the ground, and the effect of the virtual character's response to the environment was examined based on the changes in the ball as well as the body.

As shown in Fig. 24, the ejection of different parts of the sphere generated different responses. When the upper limb was ejected, the reaction was stronger, and the changes in the upper limb when the large ball was ejected were larger than those of the small ball. Due to the large mass of the lower limbs and trunk, these parts can withstand larger ejection forces, so there is no excessive action response when being ejected. The above results confirm that while our virtual character controls action generation through kinematics, it can still respond to external forces and produce varying effects based on the magnitude of these forces on different body parts. This natural responsiveness enhances the character's interaction quality, indicating that our virtual character achieves commendable physical interactivity.

To further assess the adaptability and interactivity of our character in response to environmental changes, we subjected the character to walking on constantly changing steps and being struck by a small ball (refer to the accompanying video for details).

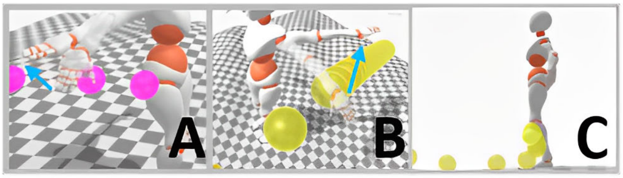


FIGURE 24. The body hit by a ball. (A)The hand hit by a small ball. (B) The hand hit by a large ball. (C) The foot hit by a large ball.

The character's walking motion is a programmed animation, where we control only the extremities and the hip, while the remaining parts adopt natural postures through the application of the elastic constraint method. The terrain on which the character moves consists of steps, each measuring 3m in length and 0.5m in width. Given the significant height difference, it poses a challenge for the character to traverse, hence the height gap between adjacent steps is kept under 0.4m. We introduced a random function to facilitate random height changes between 0.05m and 0.4m for adjacent steps, thereby creating a transformed terrain. As the character navigates through the steps, a small ball with a radius of 0.1m and a mass of 10 is propelled at the character with an acceleration of 200. The time before and after the character is hit by the small ball is 1.2 seconds, and frames are captured from the video every 0.3 seconds and incorporated into Fig. 25.

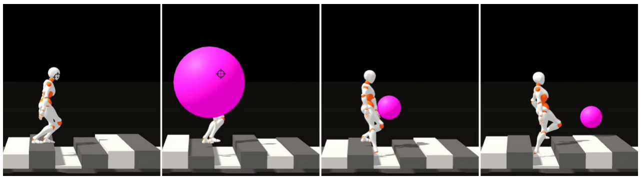


FIGURE 25. The virtual character walks on the uneven ground and is hit by a small ball.

As shown in Fig. 25, it is observed that prior to being struck by the ball, the character's body is inclined forward. After the impact, the body leans backward, and the right arm, initially swinging forward, moves backward. At this point, the character autonomously adjusts its body posture to maintain balance and continue moving forward.

The experiment reveals that our approach demonstrates robust responsiveness to changing environments. The attitude generation module, designed for environmental response, performs admirably. Whether navigating variable terrain or reacting to sudden impacts from small balls, the character responds promptly. Furthermore, our elastic constraint module optimizes the character's posture, ensuring that movements before and after collisions appear natural, free from any unnatural gestures. This indicates that our solution can significantly aid technical personnel in controlling character actions, thereby reducing the workload associated with maintaining posture rationality.

F. ADAPTABILITY EXPERIMENT UNDER STRONG INTERFERENCE ENVIRONMENT

To further assess the ability of our character to generate postures under conditions of high-speed and intense movement, we created a test environment with significant interference. We constructed the ground using four elongated blocks and introduced random fluctuations to disrupt the character's movements. Each block measures 2.4m in length, and 0.6m in both width and height, with the fluctuation amplitude ranging from 0 to 0.4m. To simulate varying intensities of movement, we established three different single-frame fluctuation amplitudes for the blocks - 0.01 m, 0.05 m, and 0.1 m, capturing one frame every 5 seconds and sampling three times within each fluctuation cycle. Additionally, we configured the observation camera to use orthographic projection, thereby eliminating any influence from the depth of field.

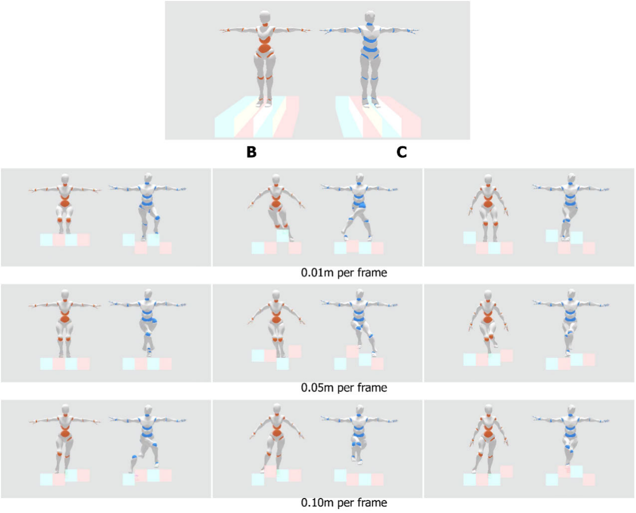


FIGURE 26. Character's motion feedback under different strong interference environments.

In Fig. 26, Character B represents the application of our proposed solution, while Character C signifies the conventional physics-based character. We restricted the movement of the hip position in both characters B and C, and exerted a downward force on their lower limbs to maintain an upright posture initially. We then induced passive movements in the characters by simulating external interference through the oscillation of the elongated blocks. As depicted in Fig.26, Character C tends to exhibit limb deformation or even disintegration during both low and high-intensity movements. However, our Character B, despite having a physical torso hinge constraint similar to Character C, was able to consistently generate normal motion postures without any deformation or limb disintegration under varying intensities of external interference. This is due to the integration of Inverse Kinematics (IK) technology in the posture calculation of our character.

Our posture correction module, which is designed based on the FABRIK-ER algorithm, strengthens the stability of the character's body part hinges. This effectively circumvents the issue of limbs easily disconnecting, a problem often encountered when characters rely solely on physical hinges. Concurrently, we imposed a force on both characters B and C to maintain their lower limbs in an upright position. Under the vibrational interference of the ground blocks, character C tends to generate irrational movements. The hip, knee, and ankle joints all display torque phenomena inconsistent with human anatomy. In contrast, our character B can generate sensible movements regardless of the interference intensity, and its overall posture consistently exhibits a commendable anthropomorphic effect throughout the interference process. This demonstrates that our elastic constraint performance can sustain excellent performance even under harsh interference environments, thereby aiding the virtual human in generating convincingly anthropomorphic postures. Moreover, when the block vibration reaches 0.10m per frame, both character B and character C will experience ground penetration phenomena. Through careful observation and repeated validation, we discovered that when only one virtual human is present in the experimental scene, neither character B nor C will penetrate the ground, even if the interference intensity exceeds 0.10m per frame. This suggests that the penetration phenomenon is related to computer performance and is not attributable to the virtual human body framework. However, due to space constraints, this topic will not be further explored in this paper.

G. DISCUSSION

The physical representation of virtual worlds has always been one of the challenges in the field of computer animation. In this study, kinematics and dynamics are combined to construct an interaction model that has good adaptability to the environment, as well as to the physical realism. The constructed model represents a lightweight physical model that requires only a small quantity of data to achieve motion control. This study thus validates the proposed elastic constraint method. The numerical changes between the joints under elastic constraint control are compared with the joints obtained from real human data acquired in the cross section, coronal plane, and sagittal plane of medical imaging. The comparison indicate that a virtual character's joints under elastic constraint control can present realistic human-like motion effects even without data input due to the elastic constraints.

In terms of motion presentation, it is shown that regardless of whether virtual characters are driven by real human motion data or keyframe animation, they can adjust their movements according to environmental changes and achieve human-like effects, which provides great assistance for computer animation design. In addition, the ball projection experiment shows that the proposed lightweight, physical, virtual character can interact with the environment, and because kinematics methods are used for motion control, there is no need for complex physical driving algorithms. In the entire scheme, artificial intelligence methods are used, and the optimized IK algorithm is combined with lightweight physical components of geometric topology to achieve realistic physical interaction effects based on simple motion control. This is due to the proposed scheme's human-like treatment of movements through elastic constraints; changes in movements affect the initial values of the motion data, allowing the characters to modify their movements in real-time based on the current motion data. Finally, in terms of reducing computing power, the proposed scheme is undoubtedly a good choice.

V. CONCLUSION

This study proposes an underlying framework that provides a high-quality underlying scheme for character animation, under which, the motion control can make a character have a good animation presentation. A practical solution is examined by combining dynamics and kinematics, which introduces an innovative idea to the character animation design.

However, the proposed model also has certain limitations. First, data obtained through motion capture technology need to be preprocessed to convert it into the form of force. The proposed processing method is not perfect, and the phenomenon of model penetration might occur when the virtual character walks or punches fast. Therefore, to enable virtual characters introduced in this study to generate more reasonable actions on their own, artificial intelligence methods could be used to train a set of virtual human behavior control strategies. Future work could further optimize the proposed model and establish an independent interactive model system.

ACKNOWLEDGMENT

The authors would like to thank LetPub (www.letpub.com) for its linguistic assistance during the preparation of this manuscript.

REFERENCES

[1] X. B. Peng, Z. Ma, P. Abbeel, S. Levine, and A. Kanazawa, "AMP: Adversarial motion priors for stylized physics-based character control," *ACM Trans. Graph.*, vol. 40, no. 4, pp. 1–20, Aug. 2021, doi: 10.1145/3450626.3459670.

[2] Y. Chen, M. Li, L. Lan, H. Su, Y. Yang, and C. Jiang, "A unified Newton barrier method for multibody dynamics," *ACM Trans. Graph.*, vol. 41, no. 4, pp. 1–14, Jul. 2022, doi: 10.1145/3528223.3530076.

[3] E. Alvarado, D. Rohmer, and M. Cani, "Generating upper-body motion for real-time characters making their way through dynamic environments," *Comput. Graph. Forum*, vol. 41, no. 8, pp. 169–181, Dec. 2022, doi: 10.1111/cgf.14633.

[4] H. Eom, D. Han, J. S. Shin, and J. Noh, "Model predictive control with a visuomotor system for physics-based character animation," *ACM Trans. Graph.*, vol. 39, no. 1, pp. 1–11, Feb. 2020, doi: 10.1145/3360905.

[5] S. Clavet, "Motion matching and the road to next-gen animation," in *Proc. GDC*, 2016, vol. 2, no. 4, p. 9.

[6] D. Holden, T. Komura, and J. Saito, "Phase-functioned neural networks for character control," *ACM Trans. Graph.*, vol. 36, no. 4, pp. 1–13, Aug. 2017.

[7] J. Juravsky, Y. Guo, S. Fidler, and X. B. Peng, "PADL: Language-directed physics-based character control," in *Proc. SIGGRAPH Asia Conf. Papers*, Nov. 2022, pp. 1–9.

[8] X. B. Peng, Y. Guo, L. Halper, S. Levine, and S. Fidler, "ASE: Large-scale reusable adversarial skill embeddings for physically simulated characters," *ACM Trans. Graph.*, vol. 41, no. 4, pp. 1–17, Jul. 2022.

[9] P. Xu, X. Shang, V. Zordan, and I. Karamouzas, "Composite motion learning with task control," *ACM Trans. Graph.*, vol. 42, no. 4, pp. 1–16, Aug. 2023.

[10] P. Xu, K. Xie, S. Andrews, P. G. Kry, M. Neff, M. Mcguire, I. Karamouzas, and V. Zordan, "AdaptNet: Policy adaptation for physics-based character control," *ACM Trans. Graph.*, vol. 42, no. 6, pp. 1–17, Dec. 2023.

[11] Q. Zhu, H. Zhang, M. Lan, and L. Han, "Neural categorical priors for physics-based character control," *ACM Trans. Graph.*, vol. 42, no. 6, pp. 1–16, Dec. 2023.

[12] J. Llobera and C. Charbonnier, "Physics-based character animation and human motor control," *Phys. Life Rev.*, vol. 46, pp. 190–219, Sep. 2023.

[13] F. Guenzkofer, H. Bubb, and K. Bengler, "Maximum elbow joint torques for digital human models," *Int. J. Hum. Factors Model. Simul.*, vol. 3, no. 2, p. 109, 2012.

[14] Z. Bhatti, I. A. Ismaili, S. Zardari, H. A. M. Malik, and M. Karbasi, "Procedural animation of 3D humanoid characters using trigonometric expressions," *Bahria Univ. J. Inf. Commun. Technol.*, vol. 9, no. 2, pp. 1–6, 2016.

[15] V. Thomasset, S. Caron, and V. Weistroffer, "Lower body control of a semi-autonomous avatar in virtual reality: Balance and locomotion of a 3D bipedal model," in *Proc. 25th ACM Symp. Virtual Reality Softw. Technol.*, Nov. 2019, pp. 1–11.

[16] D. Holden, O. Kanoun, M. Perepichka, and T. Popa, "Learned motion matching," *ACM Trans. Graph.*, vol. 39, no. 4, pp. 1–53, Aug. 2020.

[17] M. Tournier, M. Nesme, B. Gilles, and F. Faure, "Stable constrained dynamics," *ACM Trans. Graph.*, vol. 34, no. 4, pp. 1–10, Jul. 2015.

[18] A. Shapiro, D. Chu, B. Allen, and P. Faloutsos, "A dynamic controller toolkit," in *Proc. ACM SIGGRAPH Symp. Video games*, Aug. 2007, pp. 15–20.

[19] S. Kajita, F. Kanehiro, K. Kaneko, K. Yokoi, and H. Hirukawa, "The 3D linear inverted pendulum mode: A simple modeling for a biped walking pattern generation," in *Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst., Expanding Societal Role Robot. Next Millennium*, vol. 1, Jun. 2001, pp. 239–246.

[20] P. Faloutsos, M. van de Panne, and D. Terzopoulos, "Composable controllers for physics-based character animation," in *Proc. 28th Annu. Conf. Comput. Graph. Interact. Techn.*, Aug. 2001, pp. 251–260.

[21] V. B. Zordan and J. K. Hodgins, "Motion capture-driven simulations that hit and react," in *Proc. ACM SIGGRAPH/Eurograp. Symp. Comput. animation*, Jul. 2002, pp. 89–96.

[22] V. B. Zordan, A. Majkowska, B. Chiu, and M. Fast, "Dynamic response for motion capture animation," *ACM Trans. Graph.*, vol. 24, no. 3, pp. 697–701, Jul. 2005.

[23] O. Arikan, D. A. Forsyth, and J. F. O'Brien, "Pushing people around," in *Proc. ACM SIGGRAPH/Eurographics Symp. Comput. Animation*, Jul. 2005, pp. 59–66.

[24] A. Rennuit, A. Micaelli, X. Merlhiot, C. Andriot, F. Guillaume, N. Chevassus, D. Chablat, and P. Chedmail, "Balanced virtual humans interacting with their environment," 2007, *arXiv:0707.3562*.

[25] B. Stephens, "Humanoid push recovery," in *Proc. 7th IEEE-RAS Int. Conf. Humanoid Robots*, Nov. 2007, pp. 589–595.

[26] V. Zordan, A. Macchietto, J. Medina, M. Soriano, and C.-C. Wu, "Interactive dynamic response for games," in *Proc. ACM SIGGRAPH Symp. Video Games*, Aug. 2007, pp. 9–14.

[27] B. Kenwright, R. Davison, and G. Morgan, "Dynamic balancing and walking for real-time 3D characters," in *Proc. Int. Conf. Motion Games*. Cham, Switzerland: Springer, 2011, pp. 63–73.

[28] C. K. Liu, A. Hertzmann, and Z. Popović, "Learning physics-based motion style with nonlinear inverse optimization," *ACM Trans. Graph.*, vol. 24, no. 3, pp. 1071–1081, Jul. 2005.

[29] S. Jain, Y. Ye, and C. K. Liu, "Optimization-based interactive motion synthesis," *ACM Trans. Graph.*, vol. 28, no. 1, pp. 1–12, Jan. 2009.

[30] S. Ha, Y. Ye, and C. K. Liu, "Falling and landing motion control for character animation," *ACM Trans. Graph.*, vol. 31, no. 6, pp. 1–9, Nov. 2012.

[31] P. Hämäläinen, S. Eriksson, E. Tanskanen, V. Kyrki, and J. Lehtinen, "Online motion synthesis using sequential Monte Carlo," *ACM Trans. Graph.*, vol. 33, no. 4, pp. 1–12, Jul. 2014.

[32] J. Tan, K. Liu, and G. Turk, "Stable proportional-derivative controllers," *IEEE Comput. Graph. Appl.*, vol. 31, no. 4, pp. 34–44, Jul. 2011.

[33] X. B. Peng, P. Abbeel, S. Levine, and M. van de Panne, "DeepMimic: Example-guided deep reinforcement learning of physics-based character skills," *ACM Trans. Graph.*, vol. 37, no. 4, pp. 1–14, Aug. 2018.

[34] K. Bergamin, S. Clavet, D. Holden, and J. R. Forbes, "DReCon: Data-driven responsive control of physics-based characters," *ACM Trans. Graph.*, vol. 38, no. 6, pp. 1–11, Dec. 2019.

[35] J. Won and J. Lee, "Learning body shape variation in physics-based characters," *ACM Trans. Graph.*, vol. 38, no. 6, pp. 1–12, Dec. 2019.

[36] S. Andrews, I. Huerta, T. Komura, L. Sigal, and K. Mitchell, "Real-time physics-based motion capture with sparse sensors," in *Proc. 13th Eur. Conf. Vis. Media Prod. (CVMP)*, Dec. 2016, pp. 1–10.

[37] B. Kenwright, "Real-time physics-based fight characters," *Commun. Article*, pp. 1–7, Sep. 2012.

[38] A. Aristidou, J. Lasenby, Y. Chrysanthou, and A. Shamir, "Inverse kinematics techniques in computer graphics: A survey," *Comput. Graph. Forum*, vol. 37, no. 6, pp. 35–58, Sep. 2018.

[39] M. T. Hussein, A. S. Gafer, and E. Z. Fadhel, "Robot manipulator inverse kinematics using adaptive neuro-fuzzy inference system," *J. Eng. Sci. Technol.*, vol. 15, no. 3, pp. 1984–1998, 2020.

[40] S. Starke, N. Hendrich, and J. Zhang, "Memetic evolution for generic full-body inverse kinematics in robotics and animation," *IEEE Trans. Evol. Comput.*, vol. 23, no. 3, pp. 406–420, Jun. 2019.

[41] K. Erleben and S. Andrews, "Solving inverse kinematics using exact Hessian matrices," *Comput. Graph.*, vol. 78, pp. 1–11, Feb. 2019.

[42] L. Chen, T. Zielinska, J. Wang, and W. Ge, "Solution of an inverse kinematics problem using dual quaternions," *Int. J. Appl. Math. Comput. Sci.*, vol. 30, no. 2, pp. 351–361, 2020.

[43] B. Kenwright, "Real-time character inverse kinematics using the Gauss–Seidel iterative approximation method," 2022, *arXiv:2211.00330*.

[44] R. Villegas, J. Yang, D. Ceylan, and H. Lee, "Neural kinematic networks for unsupervised motion retargetting," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, Jun. 2018, pp. 8639–8648.

[45] J. Kim, Y. Seol, H. Kim, and T. Kwon, "Interactive character posing with efficient collision handling," *Comput. Animation Virtual Worlds*, vol. 31, no. 3, p. e192, May 2020.

[46] S. Silva, S. Sugahara, and W. Ugarte, "Neuranimation: Reactive character animations with deep neural networks," in *Proc. 17th Int. Joint Conf. Comput. Vis., Imag. Comput. Graph. Theory Appl.*, 2022, pp. 252–259.

[47] A. Aristidou and J. Lasenby, "FABRIK: A fast, iterative solver for the inverse kinematics problem," *Graph. Models*, vol. 73, no. 5, pp. 243–260, Sep. 2011.

[48] C. Lewin, M. Thorman, T. Waterson, C. Williams, and P. Willis, "Rod constraints for simplified ragdolls," in *Proc. 12th ACM SIGGRAPH/Eurograp. Symp. Comput. Animation*, Jul. 2013, pp. 79–84.

[49] H. Wu, J. Yu, J. Pan, and X. Pei, "A novel obstacle avoidance heuristic algorithm of continuum robot based on FABRIK," *Sci. China Technol. Sci.*, vol. 65, no. 12, pp. 2952–2966, Dec. 2022.

[50] S. Tao, H. Tao, and Y. Yang, "Extending FABRIK with obstacle avoidance for solving the inverse kinematics problem," *J. Robot.*, vol. 2021, pp. 1–10, Apr. 2021.

[51] A. Aristidou, Y. Chrysanthou, and J. Lasenby, "Extending FABRIK with model constraints," *Comput. Animation Virtual Worlds*, vol. 27, no. 1, pp. 35–57, Jan. 2016.

[52] M. Lamb, S. Lee, E. Billing, D. Högberg, and J. Yang, "Forward and backward reaching inverse kinematics (FABRIK) solver for DHM: A pilot study," in *Proc. 7th Int. Digit. Human Model. Symp.* Iowa City, IA, USA: University of Iowa, Aug. 2022, vol. 7, no. 1.

[53] G. Dong, P. Huang, Y. Wang, and R. Li, "A modified forward and backward reaching inverse kinematics based incremental control for space manipulators," *Chin. J. Aeronaut.*, vol. 35, no. 12, pp. 287–295, Dec. 2022.

[54] Z. Yin and K. Yin, "Linear time stable PD controllers for physics-based character animation," *Comput. Graph. Forum*, vol. 39, no. 8, pp. 191–200, Dec. 2020.

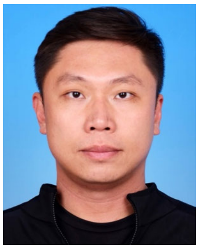
[55] S. Hong, D. Han, K. Cho, J. S. Shin, and J. Noh, "Physics-based full-body soccer motion control for dribbling and shooting," *ACM Trans. Graph.*, vol. 38, no. 4, pp. 1–12, Aug. 2019.

[56] K. Lee, S. Lee, and J. Lee, "Interactive character animation by learning multi-objective control," *ACM Trans. Graph.*, vol. 37, no. 6, pp. 1–10, Dec. 2018.

[57] D. Han, H. Eom, J. Noh, and J. S. Shin (formerly Sung Yong Shin), "Data-guided model predictive control based on smoothed contact dynamics," *Comput. Graph. Forum*, vol. 35, no. 2, pp. 533–543, May 2016.

[58] S. Carensac, N. Pronost, and S. Bouakaz, "Physics-based control of walking virtual characters in low frequency simulations," in *Proc. 31st Int. Conf. Comput. Animation Social Agents*, May 2018, pp. 77–82.

[59] C. R. Booher and B. S. Goldsberry, "NASA's man-systems integration standards: A human factors engineering standard for everyone in the nineties," in *Proc. Dual-Use Space Technol. Transf. Conf. Exhib.*, vol. 1, 1994, pp. 211–220.



BIN HU was born in Shandong, China. He received the Ph.D. degree in interaction and visual design from the Faculty of Creative Industries, Queensland University of Technology, Brisbane, Australia. He is currently an Assistant Professor with the Faculty of Humanities and Arts, Macau University of Science and Technology. His research interests include user experience and empathic and visual communication designs.

HUI GUO received the M.Sc. degree from the School of Electronic and Information Engineering, Guangxi Normal University, Guilin, China, in 2010. She is currently pursuing the Ph.D. degree in digital media with the Faculty of Arts and Humanities, Macau University of Science and Technology, Macau, China. She is also a Professor in electronic information engineering with Wuzhou University, Wuzhou, China. Her research interests include video image processing, data visualization, and networking and applications.

YULING YANG received the bachelor's degree in engineering. She is currently pursuing the master's degree in game design with the Faculty of Humanities, Macau University of Science and Technology, China. She is also a Research Assistant with Guangxi Key Laboratory of Machine Vision and Intelligent Control. Her research interests include physics-based character and data visualization and applications.

XIONGJIE TAO received the bachelor's degree in engineering. He is currently pursuing the master's degree in game design with the Faculty of Arts and Humanities, Macau University of Science and Technology, China. He was an Assistant Engineer in IoT cloud engineering, in 2020, a Software Designer, in 2021, and Guangdong Province, the number of intellectualization experts think tank experts title, in 2024. His research interests include digital twin and the IoT cloud computing and industrial IoT.

JIE HE was born in Yiyang, Hunan, China. He received the M.Sc. degree from Guangxi Normal University, Guilin, Guangxi, China. He is currently pursuing the Ph.D. degree with the College of Computer Science and Electronic Engineering, Hunan University, Changsha, Hunan. Since 2020, he has been the Executive Deputy Director of Guangxi Key Laboratory of Machine Vision and Intelligent Control. His current research interests include image processing, data visualization, and medical image data mining.